

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Vizualizace rozvrhu učebny na e-ink displayi

Visualization of Classroom Schedule on E-ink Display

Zadání diplomové práce

Student:

Bc. Viktor Jakš

Studijní program:

N2647 Informační a komunikační technologie

Studijní obor:

2612T025 Informatika a výpočetní technika

Téma:

Vizualizace rozvrhu učebny na e-ink displayi
Visualization of Classroom Schedule on E-ink Display

Jazyk vypracování:

čeština

Zásady pro vypracování:

Diplomová práce bude navazovat na semestrální projekt, jehož náplní bylo použití vývojové desky ESP32 k vykreslení webových stránek na e-ink displayi. Hlavním cílem této práce bude zobrazování rozvrhů jednotlivých učeben získávaných z univerzitního serveru. Vývoj systému bude směřovat k co nejnižším nárokům na spotřebu energie, aby byl schopen dlouhodobého provozu na baterie.

1. Pro vývoj použijte dodaný e-ink display a vývojovou desku ESP32. Obojí stručně popište.
2. Rozšiřte projekt o možnost zobrazování rozvrhů hodin. Vykreslení bude optimalizované pro daný display, aby byl obsah co nejlépe čitelný. Obsah rozvrhu bude získáván z univerzitního serveru pomocí rozhraní INBUS. Toto rozhraní stručně popište.
3. Systém koncipujte tak, aby bylo dosaženo minimální spotřeby elektrické energie.
4. Přidejte možnost přenosu dat pomocí Bluetooth a srovnajte spotřebu energie v porovnání s použitím WiFi.
5. Proveďte měření spotřeby energie. Zaměřte se na spotřebu při jednom vykreslovacím cyklu a při jednotlivých fázích vykreslení. Prozkoumejte možnosti běhu systému na baterie.
6. K vývoji využijte oficiální vývojový framework ESP-IDF a programovací jazyk C/C++. Výsledný zdrojový kód programu řádně okomentujte.

Seznam doporučené odborné literatury:

- [1] ESP32 Hardware Design Guidelines: ETSI, 37 s. Dostupné z WWW: <https://www.espressif.com/sites/default/files/documentation/esp32_hardware_design_guidelines_en.pdf>.
- [2] Driving Electronic Paper Displays (E-paper), AN0063 - Application Note; Dostupné z WWW: <https://www.silabs.com/documents/public/application-notes/AN0063.pdf>
- [3] Free RTOS Reference Manual, Dostupné z WWW: https://www.freertos.org/Documentation/FreeRTOS_Reference_Manual_V9.0.0.pdf


Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. David Seidl, Ph.D.**

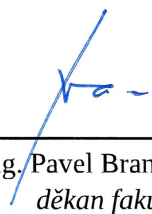
Datum zadání: 01.09.2019

Datum odevzdání: 30.04.2020





doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry



prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární
prameny a publikace, ze kterých jsem čerpal.

V Ostravě 12. května 2020

Jakub Vihlston
.....

Rád bych poděkoval Ing. Davidu Seidlovi, Ph.D. za všechny odborné rady, cenné připomínky, ochotu a vstřícný přístup při vypracovávání této diplomové práce. Dále bych chtěl poděkovat Ing. Radimu Dostálovi za odbornou pomoc týkající se univerzitního informačního systému.

Abstrakt

Diplomová práce navazuje na semestrální projekt, v rámci kterého byl vytvořen systém pro zobrazování HTML stránek na e-ink display pomocí vývojové desky ESP32. Cílem této diplomové práce je rozšíření semestrálního projektu o možnost vykreslování rozvrhů hodin na základě dat poskytnutých univerzitním informačním systémem. Vývoj je zaměřen na dobrou čitelnost obsahu vykreslovaného na e-ink display a nízkou spotřebu elektrické energie. Součástí práce je měření spotřeby energie kvůli možnosti provozu zařízení na baterie. Vytvořený systém je rozdělen na dvě hlavní části. První je klientská část, která vznikla pomocí vývojového frameworku ESP-IDF a programovacího jazyka C. Druhou částí je server, který je založen na programovacím jazyce Python a webovém frameworku Django.

Klíčová slova: E-ink display, ESP32, ESP-IDF, C, Python, Django, Bluetooth, WiFi

Abstract

The master thesis follows up on a semestral project in which was created a system for displaying HTML webpages on e-ink display using development board ESP32. The goal of this master thesis is to extend the semestral project by the possibility of drawing school schedules based on the data provided by university information system. Development is focused on good readability of the content drawn on the e-ink display and on low power consumption. Measuring of the power consumption is done because of the possibility to run the devices on batteries. The created system is divided into two main parts. The first part is client which was created using the development framework ESP-IDF and the programming language C. The second part is server which is based on the programming language Python and the web framework Django.

Keywords: E-ink display, ESP32, ESP-IDF, C, Python, Django, Bluetooth, WiFi

Obsah

Seznam použitých zkratk a symbolů	9
Seznam obrázků	11
Seznam tabulek	12
Seznam výpisů zdrojového kódu	13
1 Úvod	14
2 Zvolený postup	16
2.1 Zobrazování webových stránek	16
2.2 Zobrazování rozvrhů hodin	18
3 Použitý hardware	19
3.1 Server	19
3.2 Vývojová deska	19
3.3 E-ink display	20
3.4 Propojení modulu ESP32 a e-ink displaye	22
4 Instalace softwaru	23
4.1 Serverová část	23
4.2 Klientská část	26
5 Webová aplikace	27
5.1 Možnosti konfigurace	27
5.2 Zobrazování přehledů	28
5.3 Přístup k API	29
5.4 Spuštění webového serveru	29
6 Vykreslovací aplikace	30
6.1 Snímky webových stránek	30
6.2 Data rozvrhů hodin	35
6.3 Spuštění vykreslovací aplikace	43
7 Klientská aplikace	45
7.1 Společná část	45
7.2 Vykreslování webových stránek	48
7.3 Zpracování rozvrhů hodin	49
7.4 Spuštění klientské aplikace	54

8	Spotřeba elektrické energie	56
8.1	Režimy spotřeby elektrické energie	56
8.2	Spotřeba modulu ESP32 v různých situacích	58
8.3	Bezdrátový přenos dat	59
8.4	Vykreslení obsahu na e-ink display	61
8.5	Souhrnné informace o spotřebě elektrické energie	63
8.6	Zhodnocení spotřeby elektrické energie	64
8.7	Výběr vhodných baterií	65
9	Závěr	67
	Literatura	68

Seznam použitých zkratk a symbolů

API	– Application Programming Interface
ASCII	– American Standard Code For Information Interchange
B	– Byte
BLE	– Bluetooth Low Energy
CEF	– Chromium Embedded Framework
CPU	– Central Processing Unit
CSS	– Cascading Style Sheets
dpi	– Dots Per Inch
DVD	– Digital Versatile Disc
e-ink	– Electronic Ink
ESP-IDF	– Espressif IoT Development Framework
GHz	– Gigahertz
GPIO	– General-Purpose Input/Output
H	– Hloubka
HTML	– Hypertext Markup Language
HTTP	– Hypertext Transfer Protocol
HTTPS	– Hypertext Transfer Protocol Secure
I2C	– Inter-Integrated Circuit
I2S	– Inter-IC Sound
IC	– Integrated Circuit
ID	– Identifikátor
INBUS	– Integrovaná Brána Univerzitních Systémů
IOT	– Internet Of Things
IP	– Internet Protocol
ITU	– International Telecommunication Union
ITU-R	– ITU Radiocommunication Sector
JSON	– JavaScript Object Notation
KiB	– Kibibyte
mA	– Miliampér
mAh	– Miliampérhodina
mAs	– Miliampérsekunda
Mbit	– Megabit
MHz	– Megahertz
MiB	– Mebibyte
MISO	– Master Input / Slave Output
mm	– Milimetr

MOSFET	– Metal Oxide Semiconductor Field Effect Transistor
MOSI	– Master Output / Slave Input
ms	– Milisekunda
mW	– Miliwatt
PNG	– Portable Network Graphics
PSK	– Pre-Shared Key
px	– Pixel
REST	– Representational State Transfer
RTC	– Real-Time Clock
s	– Sekunda
SCLK	– Serial Clock
SPI	– Serial Peripheral Interface
SQL	– Structured Query Language
SRAM	– Static Random Access Memory
SS	– Slave Select
Š	– Šířka
TCP	– Transmission Control Protocol
ULP	– Ultra Low Power
URL	– Uniform Resource Locator
USB	– Universal Serial Bus
UTF-8	– 8-bit Unicode Transformation Format
UUID	– Universally Unique Identifier
V	– Volt
venv	– Virtual Environment
WPA2	– Wi-Fi Protected Access II
XML	– Extensible Markup Language
Xvfb	– X Virtual Framebuffer
μA	– Mikroampér
°C	– Stupeň Celsia

Seznam obrázků

1	Čtečka elektronických knih	14
2	Flashdisk s e-ink displayem	14
3	Buňky e-ink displaye	21
4	Struktura souboru s rozvrhem hodin	42
5	Rozložení e-ink displaye	46

Seznam tabulek

1	Použité piny pro propojení modulu ESP32 a e-ink displaye	22
2	Odebíraný proud modulem ESP32 v různých situacích	58
3	Srovnání spotřeby elektrické energie u WiFi a BLE	61
4	Spotřeba elektrické energie v jednotlivých fázích vykreslování obsahu	62
5	Rozdělení spotřeby elektrické energie při vykreslování obsahu	62
6	Celková spotřeba elektrické energie	63
7	Rozdělení celkové spotřeby elektrické energie	63
8	Možná úspora elektrické energie při použití BLE místo WiFi	64

Seznam výpisů zdrojového kódu

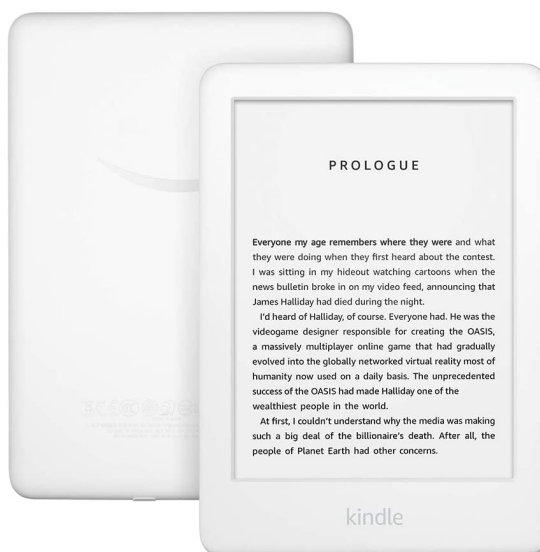
1	Aktualizace databáze balíků	23
2	Kontrola verze programovacího jazyka Python	24
3	Instalace programovacího jazyka Python	24
4	Instalace správce balíků pip	24
5	Instalace modulu venv pro vytváření virtuálních prostředí	24
6	Vytvoření složky s virtuálním prostředím	24
7	Aktivace virtuálního prostředí	25
8	Instalace knihovny Pillow a nástroje CEF-Python	25
9	Instalace webového frameworku Django	25
10	Aktualizace webového frameworku Django	25
11	Instalace požadovaných balíků pro ESP-IDF	26
12	Stažení balíku nástrojů pro ESP-IDF	26
13	Stažení vývojového frameworku ESP-IDF	26
14	Nastavení cest ke složkám ESP-IDF	26
15	Instalace balíků pro Python, které ESP-IDF vyžaduje	26
16	Spuštění webového serveru	29
17	Spuštění vykreslovací aplikace	43

1 Úvod

S běžně používanými zobrazovacími technologiemi se lze setkat především u televizí, počítačových monitorů a obrazovek mobilních telefonů. Všechny je spojuje stejná vlastnost, kterou je neustálá spotřeba elektrické energie pro zobrazování obsahu. Právě v tom se liší technologie e-ink. Pouze malé množství energie je potřeba pro vykreslení statického obrazu, který zůstává libovolnou dobu nezměněn i po odpojení od zdroje napájení.

Displeje založené na technologii e-ink mají díky svým vlastnostem širokou oblast využití. Nasazení je vhodné především v případech, kdy je prioritou energetická nenáročnost, dobrá viditelnost obrazu a obsah, který se příliš často nemění. Mezi nedostatky této technologie patří delší doba překreslení a absence barevných možností, které bývají k dispozici u jiných zobrazovacích technologiích.

Příkladem použití mohou být například cedulky v policích obchodů pro zobrazování informací o zboží, jako je cena, datum trvanlivosti a čárový kód. Dále se může jednat o informační cedule s jízdními řády a zpožděními spojů na zastávkách veřejné dopravy, čtečky elektronických knih (viz obrázek 1), informace na platebních kartách nebo drobné displeje obsahující stav zařízení, jako je například dostupné místo na flashdisku (viz obrázek 2).



Obrázek 1: Čtečka elektronických knih [1]



Obrázek 2: Flashdisk s e-ink displayem [2]

Pro obecné použití e-ink displaye je potřeba mít k dispozici způsob, jak na něj vykreslit libovolný obsah. Za tímto účelem vznikl semestrální projekt, na který tato diplomová práce navazuje. V rámci semestrálního projektu proběhl výběr vhodného e-ink displaye a vývojové desky, která je potřeba k přenesení obsahu na připojený display. Poté byl vytvořen systém, díky kterému je možné na e-ink display vykreslit snímek libovolné webové stránky.

Diplomová práce přidá možnost zobrazování rozvrhů hodin pomocí vybraného e-ink displaye. Data rozvrhů hodin budou získávána z univerzitního informačního systému v textové podobě. Vykreslování na display bude probíhat ve formě tabulky obsahující informace pro aktuální týden. Nebudou tak zahrnuty nepotřebné informace, jako jsou například hodiny studentů kombinovaného studia, které se konají v jiných týdnech. Při opakovaném vykreslování budou také promítnuty změny v rozvrhu hodin, které nastanou až v průběhu semestru.

Použitím většího množství e-ink displayů by mohly být nahrazeny papírové rozvrhy hodin, které bývají umístěny u učeben. V takovém případě musí být možný provoz na baterie, jelikož není k dispozici jiný zdroj napájení. Z toho důvodu bude systém vyvíjen takovým způsobem, aby byla zajištěna co nejmenší energetická náročnost. Pro zjištění možné doby provozu na baterie budou použity informace z provedeného měření spotřeby elektrické energie.

V následující kapitole bude popsán postup zvolený pro vypracování této práce. Další dvě kapitoly se budou věnovat použitému hardwaru a instalaci softwaru. Následovat budou tři kapitoly zabývající se jednotlivými částmi softwaru vytvořenému v rámci této práce. Předposlední kapitola bude obsahovat informace týkající se spotřeby elektrické energie. V závěrečné kapitole bude zhodnocen stav vytvořené práce a popsána možnost dalšího vývoje.

2 Zvolený postup

Projekt je rozdělen na dvě hlavní části. První z nich je klient, druhou je sever. Klientská část se skládá z vývojové desky a připojeného e-ink displaye. Klient provádí komunikaci se serverem, který poskytuje data pro vykreslení na e-ink display. Server také zpřístupňuje webové rozhraní, které uživateli umožňuje zobrazování přehledů a vytváření konfigurace.

Vykreslovaný obsah se dělí na webové stránky a rozvrhy hodin. Oba tyto typy obsahu se zcela liší jak v jejich získávání, tak v následném zpracování. Pro každý z nich je tedy potřeba určit, jakým způsobem postupovat.

2.1 Zobrazování webových stránek

Pro vykreslení obrazu na základě HTML kódu webové stránky existují dva možné způsoby, jak postupovat. První z nich je navštívit webovou stránku pomocí klientského zařízení připojeného k e-ink displayi, stáhnout HTML kód a na jeho základě vytvořit obraz webové stránky pro vykreslení na e-ink display. Druhý způsob je provést vytvoření obrázku webové stránky na serveru a předat ho klientovi k vykreslení. Zvolena byla druhá možnost, a to z následujících důvodů:

- S pomocí serveru je na e-ink displayi možné zobrazit jakoukoliv webovou stránku. Místo manuální implementace několika nejběžnějších HTML značek na straně klienta je na serveru provozován plnohodnotný webový prohlížeč. Ten je použit k navštěvování a vykreslování nakonfigurovaných webových stránek. Výsledkem je vykreslení takového obsahu na e-ink display, který by uživatel viděl při použití běžného webového prohlížeče.
- Využití webového prohlížeče zajistí správné načtení a použití součástí, kterými jsou například *CSS* a *Javascript*. V dnešní době se jedná o nutnost, jelikož je na těchto technologiích založena naprostá většina moderních webových stránek. [3, 4]
- Obrazová data jsou vždy připravena předem, aby byl možný okamžitý přenos ve chvíli, kdy jsou data klientem vyžádána ke stažení. Není tak potřeba čekat na zdlouhavý proces, při kterém musí server navštívit webovou stránku a stáhnout všechna její data, vytvořit snímek stránky ve formě obrázku a převést ho do formátu, který je vhodný pro klienta a jeho e-ink display. Tohle přednačtení obsahu zkracuje dobu, kterou musí klient strávit v aktivním režimu, což vede ke snížení spotřeby elektrické energie.
- Na straně klienta není potřeba téměř žádné zpracování získávaných dat. Stahovaný soubor obsahuje obrazovou bitmapu s barvami v odstínech šedi. Jednotlivé pixely jsou po osmi shlukovány do společného byte takovým způsobem, jakým to vyžaduje e-ink display. Data jsou při stahování umisťována do datových polí, která jsou poté přímo předána e-ink displayi k vykreslení. Není tak ztracen čas zbytečnými operacemi, jako je například převod z jiného obrazového formátu.

- Vlastní server je možné umístit takovým způsobem, aby byla zajištěna co nejlepší síťová konektivita, jako je například stejná místní síť, ve které se nachází i klient. Kromě zajištění vysoké komunikační rychlosti a nízkého zpoždění mezi serverem a klientem je tak řešen i problém přístupu k webovým stránkám pomocí pomalejšího připojení k internetu. Navíc k vhodnému síťovému připojení je možné poskytnout i dostatečně výkonné hardwarové vybavení, na kterém software serveru poběží.
- Stažení moderní webové stránky obvykle vyžaduje přenos velkého množství dat. To může být problém nejen kvůli dlouhé době stahování, ale také kvůli vysoké náročnosti na paměť u zařízení, která jí mají jen velmi omezené množství. Kromě rozsáhlého obsahu HTML kódu webových stránek jsou problémem také externí zdroje, díky kterým je načítání stránek ještě komplikovanější. Obvykle se jedná o různé typy obrázků, Javascript, CSS a soubory s fonty.
- Přestože se může zdát, že stažení a zpracování velmi malé a jednoduché webové stránky přímo klientem by mohlo být rychlejší a méně náročné na spotřebu elektrické energie, než stahování obrazové bitmapy z vlastního serveru, zlepšení by ve skutečnosti bylo téměř nulové. Důvody jsou následující:
 - Soubor s obrazovými daty má relativně malou velikost. Při použití parametrů e-ink displaye, který byl zvolen pro tuto práci, je možné vypočítat, že přenášená obrazová informace má velikost pouze 90 KiB. Při použití jiného e-ink displaye, který má odlišné rozměry nebo počet zobrazovaných odstínů šedi, může být velikost souboru větší nebo menší.
 - Stažení souboru je hotovo prakticky okamžitě. Použité klientské zařízení, kterým je modul ESP32, podporuje propustnost dat přes *TCP* spojení o rychlosti až 20 Mbit/s, takže je přenos takového množství dat přes WiFi velmi rychlý. [5]
 - Časově nejnáročnější část síťové komunikace je proces připojování k bezdrátové síti WiFi. Vytvoření spojení obvykle trvá okolo čtyř sekund. Měření tohoto času probíhalo na síti zabezpečené pomocí *WPA2-PSK*.
 - Dostatečně malých a jednoduchých webových stránek, u kterých by mohlo nastat zlepšení, není velké množství. Ve zvláštních případech by mohla být na e-ink displayi zobrazována vlastní připravená webová stránka, cílem však je možnost zobrazení jakéhokoliv dostupného webu.
- Každou webovou stránku je kvůli vykreslení potřeba navštívit pouze jednou. To může být užitečné v situaci, kdy je více klientských zařízení nastaveno k zobrazování stejného obsahu.
- Všechny změny týkající se obsahu vykreslovaného na e-ink display je možné provádět vzdáleně bez nutnosti fyzického kontaktu s klientským zařízením. Bez použití vlastního serveru by při každé změně bylo potřeba zkompileovat nový kód se změnami, připojit klientské zařízení k počítači a nahrát na něj nový program.

2.2 Zobrazování rozvrhů hodin

Pro zobrazování rozvrhů hodin je také použit vlastní server, stejně jako u zobrazování webových stránek. Důvody k jeho použití jsou následující:

- Stejně jako u vykreslování webových stránek, použití vlastního serveru umožňuje zajistit dobrou síťovou konektivitu.
- Obdobně je k dispozici vzdálené provádění změn ve vykreslovaném obsahu. Kromě vybrání jiného rozvrhu hodin, který má být na e-ink displayi zobrazován, je možné změnit obsah na vybranou webovou stránku. Stejně tak lze opět začít vykreslovat rozvrh hodin.
- Webové rozhraní umožňuje snadné vybrání místnosti, jejíž rozvrh hodin má být vykreslován. Výběr probíhá ze seznamu dostupných místností, které se nacházejí ve zvolené budově. Tuto budovu je nejprve potřeba vybrat, stejně tak jako areál, ve kterém se budova nachází. Bez tohoto rozhraní by musely být ručně vyhledávány číselné identifikátory v seznamu všech dostupných položek.
- Ve chvíli, kdy klient stáhne data ze serveru, je pro něj uložena aktuální časová informace. Ta je pro každého klienta zobrazena v přehledu, který je dostupný pomocí webového rozhraní. Je tak možné mít přehled o času a datu poslední aktualizace všech klientských zařízení.
- V případě, že by se univerzitní informační systém stal nedostupným, rozvrhy hodin uložené na vlastním serveru zůstanou nadále dostupné.
- Data rozvrhů hodin jsou připravena v takovém formátu, aby na klientském zařízení stačilo pouze namapovat struktury programovacího jazyka *C* na datové pole se staženými daty. Je tak možné k informacím přímo přistupovat bez dalšího předzpracování. To je prováděno na serveru a vypořádává se mimo jiné s parsováním dat ve formátu *JSON*, se změnou kódování textu a také s diakritikou.
- Vzhledem k použitému způsobu ověřování, které je potřeba během přístupu k univerzitnímu informačnímu systému s rozvrhovými daty, by se mohlo stát použití více klientských zařízení současně problematickým. Důvodem je získávání autorizačního tokenu, který je přidělován na základě předsdíleného hesla a po dobu platnosti je opakovaně používán. Bez použití serveru, který získává všechny požadované rozvrhy hodin, by muselo být heslo uloženo do všech klientských zařízení. Před každým stažením rozvrhových dat by pak muselo být vyžádáno přidělení nového autorizačního tokenu, čímž nastane zneplatnění předchozího autorizačního tokenu.

3 Použitý hardware

Pro realizaci projektu jsou potřeba celkem tři hardwarová zařízení. Pro klientskou část se jedná o e-ink display a vývojovou desku. Dále je potřeba počítač sloužící jako server. V této kapitole jsou tato tři hardwarová zařízení popsána.

3.1 Server

Na server nejsou kladeny žádné zvláštní požadavky. Může se jednat prakticky o jakýkoliv počítač, který je schopen provozovat webový server a internetový prohlížeč. Požadovaný výkon tohoto serveru závisí především na počtu aktivních klientů, kteří mají být obsluhováni. Server by měl být schopen odpovídat klientům co nejrychleji, aby během čekání nebylo plýtváno jejich časem a elektrickou energií.

3.2 Vývojová deska

Přestože existuje velké množství různých vývojových desek, pro tento projekt je vhodná pouze taková, která splňuje následující požadavky:

- Nízká spotřeba elektrické energie
- Možnost dlouhého spánku bez nutnosti externího zdroje probuzení
- Komunikační rozhraní kompatibilní s vybraným e-ink displayem
- Přítomnost vhodné technologie pro bezdrátovou komunikaci
- Dobrá dostupnost vybraného zařízení a jeho dokumentace

Zařízení, které splňuje všechny požadavky, je *ESP32-DevKitC* od *Espressif Systems*. Jedná se o vývojovou desku s nízkými energetickými nároky. Obsahuje modul *ESP32-WROOM-32*, který poskytuje možnost bezdrátové konektivity pomocí *WiFi* a *Bluetooth*. Základem tohoto modulu je čip *ESP32-D0WDQ6*. Důležité vlastnosti modulu jsou následující: [6, 7]

- **CPU:**
 - 32-bitový mikroprocesor Xtensa LX6
 - Frekvence nastavitelná od 80 do 240 MHz
 - Dvě jádra procesoru
- **Paměť:** 520 KiB SRAM, 4 MiB flash
- **WiFi:**
 - Podpora standardů 802.11 b/g/n (2,4 GHz)
 - Teoretická propustnost packetů až 150 Mbit/s

- **Bluetooth:**
 - Klasické i nízkoenergetické (BLE)
 - Verze specifikace 4.2
 - Podpora tříd výkonu 1, 2 a 3
- **38 pinů:** Podpora GPIO, I²C, I²S, SPI a mnoho dalšího
- **Provozní napětí:** 3,3 V (doporučený rozsah: 3,0 – 3,6 V)
- **Odebíraný proud:** 0 – 240 mA
- **Provozní teplota:** -40 – 85 °C

3.3 E-ink display

V současné době je na trhu k dispozici relativně velké množství různých e-ink displayů. Většina z nich je však nevhodná kvůli jejich malým rozměrům a malému rozlišení. E-ink display vybraný pro tento projekt je jeden z největších, které byly v době výběru dostupné. Mezi další důvody pro volbu tohoto e-ink displaye patří vhodné komunikační rozhraní a možnost vykreslování současně černou i červenou barvou. Zvolený e-ink display od výrobce *Pervasive Displays* má následující parametry: [8]

- **Velikost displaye (mm):** 251 (Š) x 204,08 (V) x 1,2 (H)
- **Aktivní oblast (mm):** 237,6 (Š) x 190,08 (V)
- **Rozlišení:** 960 x 768 px
- **Hustota pixelů:** 103 dpi
- **Počet barev:** 3 (černá, bílá, červená)
- **Pozorovací úhel:** téměř 180°
- **Komunikační rozhraní:** SPI
- **Doba překreslení:** 41 sekund
- **Provozní teplota:** 10 – 40 °C
- **Provozní napětí:** 3,0 V (doporučený rozsah: 2,3 – 3,6 V)
- **Odebíraný proud:** 24,4 – 71,3 mA
- **Spotřeba energie pro překreslení:** 1 000,4 – 2 923,3 mAs

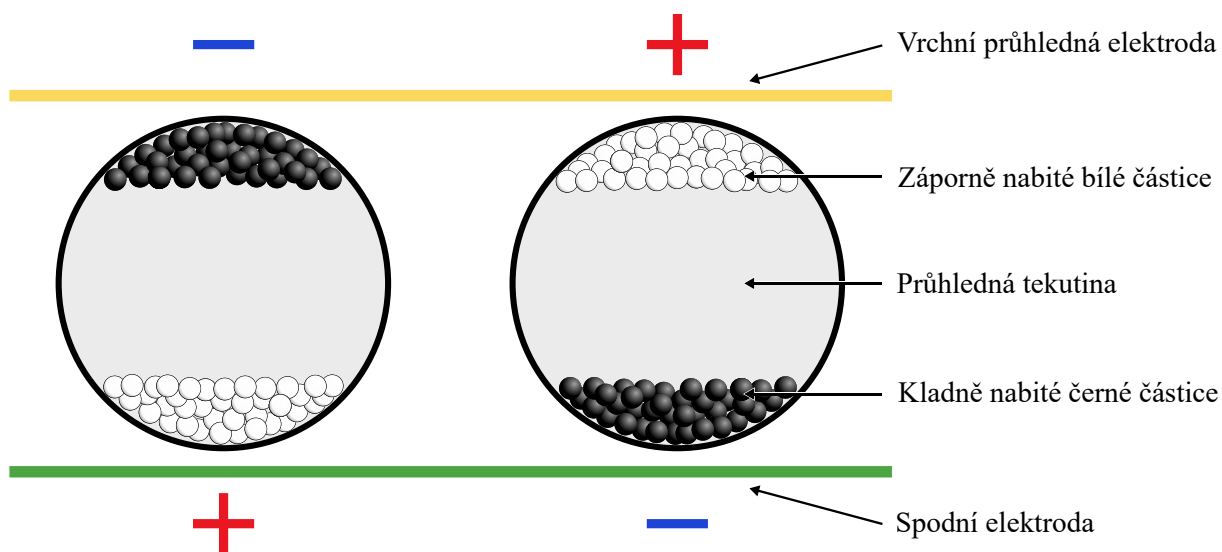
3.3.1 Technologie e-ink

Hlavní výhoda technologie e-ink je, že display nepotřebuje žádný elektrický proud k zachování vykresleného obrazu. Potřeba je pouze malé množství elektrického proudu ve chvíli, kdy je na display vykreslován nový obraz.

E-ink display je tvořen velkým množstvím drobných buněk. Každá buňka obsahuje záporně nabitě bílé částice a kladně nabitě černé částice. Tyto částice jsou ponořeny v průhledné tekutině. Když je aplikováno elektrické pole s kladným nebo záporným nábojem, odpovídající částice se přesunou na vrchní stranu buňky a stanou se tak viditelnými pro pozorovatele. Princip činnosti této technologie je znázorněn na obrázku 3, kde se levá buňka při pohledu z vrchu jeví jako černá a pravá buňka jako bílá. [9]

V současnosti běžné e-ink displaye zobrazují pouze černou a bílou, případně i třetí barvu. Jsou však dostupné i modely zobrazující odstíny šedi. Dále existují e-ink displaye zobrazující také barvy. Toho bývá obvykle dosaženo použitím barevných filtrů na buňkách e-ink displaye.

Různé e-ink displaye se liší v době, kterou trvá jedno vykreslení obrazu. Potřebná doba může být od zlomku sekundy po desítky sekund. Při výběru e-ink displaye je vhodné tento parametr sledovat, jelikož v některých případech může být příliš dlouhé vykreslování nežádoucí.



Obrázek 3: Buňky e-ink displaye [9]

3.4 Propojení modulu ESP32 a e-ink displaye

Modul ESP32 a e-ink display je potřeba propojit pomocí odpovídajících pinů. Použité piny jsou vypsány v tabulce 1 včetně názvu, podle kterého lze v dokumentaci dohledat bližší informace. Přenos dat z modulu ESP32 na e-ink display probíhá pomocí protokolu *SPI*.

Modul ESP32	E-ink display	Název pinu
3V3	1	VCC
IO12	2	GUARD2
IO18	7	SPI_CLOCK
IO27	8	BUSY
IO5	9	DATA/COMMAND
IO26	10	RESET
IO14	11	PANEL_ON
IO19	15	SPI_MOSI
GND	17	TEST/SBWTCK
IO23	19	EPD_CHIP_SELECT
GND	20	GROUND

Tabulka 1: Použité piny pro propojení modulu ESP32 a e-ink displaye [10]

3.4.1 Serial Peripheral Interface (SPI)

SPI je protokol pro komunikaci mezi propojenými zařízeními. Komunikace probíhá v jeden okamžik mezi právě dvěma zařízeními. Jedno ze zařízení je v režimu *master*, v tomto případě to je modul ESP32. Další zařízení jsou v režimu *slave*, zde se jedná o e-ink display. Master je řídicí a slave je podřízené zařízení. Master vydává instrukce, které slave přijímá. Komunikace je sériová a je řízena hodinovým signálem. K propojení zařízení slouží následující čtyři typy vodičů: [11]

- **SCLK** (Serial Clock) – Přenáší signál, který generuje master. Slouží k synchronizaci výstupu, který vytváří master se vstupem, který přijímá slave.
- **MOSI** (Master Output / Slave Input) – Vodič využívá master k odesílání dat, která přijímá slave.
- **MISO** (Master Input / Slave Output) – Slouží k přenášení dat, která odesílá slave a přijímá master.
- **SS** (Slave Select) – Jelikož může být jeden master připojen současně k více zařízením v režimu slave, je použit *SS* k výběru, se kterým zařízením má být komunikováno.

4 Instalace softwaru

Pro spuštění aplikací vytvořených v této práci musí být nainstalován potřebný software. Ten se liší pro serverovou a pro klientskou část. Jelikož jsou všechny použité nástroje multiplatformní, vývoj kódu i provoz serveru může probíhat na různých operačních systémech, jako jsou *Linux*, *Windows* a *macOS*. Funkčnost byla testována na *Windows 10* a *Ubuntu 18.04*.

Instalace veškerého potřebného softwaru je velmi jednoduchá, jelikož jsou všechny součásti jejich vydavateli dobře připraveny a zdokumentovány. Na jednotlivých operačních systémech však může instalace vypadat odlišně. Stažení softwaru probíhá z webové stránky vydavatele nebo z repozitáře poskytnutého operačním systémem.

4.1 Serverová část

Součásti, které je potřeba nainstalovat na server, jsou následující:

- Podpora programovacího jazyka *Python*
- Webový framework *Django*
- Nástroj *CEF-Python*
- Vhodný operační systém

4.1.1 Operační systém

Následující popis instalace součástí je určen pro operační systém *Ubuntu 18.04*. Pro jiné Linuxové distribuce se postup může mírně lišit. Instalaci samotného operačního systému je možné provést například stažením souboru s obrazem instalačního *DVD*.

4.1.2 Aktualizace databáze balíků

Před instalací balíků je vhodné nejprve aktualizovat jejich databázi. Je tak zajištěno, že budou k dispozici v nejnovější verzi.

```
sudo apt update
```

Výpis 1: Aktualizace databáze balíků

4.1.3 Programovací jazyk Python

Programovací jazyk *Python* je vyžadován ve verzi 3.6 nebo novější. Vzhledem k použitému operačnímu systému není jeho instalace potřeba, jelikož už je zde předinstalován. Při použití jiné Linuxové distribuce je vhodné provést kontrolu přítomné verze.

```
python3 --version
```

Výpis 2: Kontrola verze programovacího jazyka Python

V případě, že Python není nainstalován, nebo má starší verzi, lze provést jeho doinstalování.

```
sudo apt install python3 -y
```

Výpis 3: Instalace programovacího jazyka Python

4.1.4 Správce balíků pip

Nástroj *pip* je správce balíků pro programovací jazyk Python. Pomocí něj lze přidávat součásti, které poskytují další funkcionalitu. Instalaci tohoto nástroje lze přeskočit, pokud bude používáno virtuální prostředí (viz následující kroky), jelikož je *pip* automaticky jeho součástí. [12, 13]

```
sudo apt install python3-pip -y
```

Výpis 4: Instalace správce balíků pip

4.1.5 Modul venv pro vytváření virtuálních prostředí

K programovacímu jazyku Python je možné doinstalovat modul *venv*. Slouží k vytváření izolovaných virtuálních prostředí, do kterých lze instalovat balíky bez ovlivnění globální instalace Pythonu. Použití virtuálního prostředí je také bezpečnější, jelikož pak pro instalaci balíků není potřeba oprávnění uživatele *root*. Přestože virtuální prostředí není nutností, jeho instalace je doporučena. Pozdější postup spouštění serverové aplikace a následující kroky instalace předpokládají, že je virtuální prostředí nainstalováno. [13]

```
sudo apt install python3-venv -y
```

Výpis 5: Instalace modulu venv pro vytváření virtuálních prostředí

4.1.6 Vytvoření složky s virtuálním prostředím

Pomocí modulu *venv* je možné vytvořit virtuální prostředí. Jedná se o složku obsahující potřebné soubory. Ta může být vytvořena v libovolném umístění. Celé virtuální prostředí lze odstranit smazáním této složky, která je v tomto případě vytvářena v domovském adresáři. [13]

```
python3.6 -m venv ~/eink_venv
```

Výpis 6: Vytvoření složky s virtuálním prostředím

4.1.7 Použití virtuálního prostředí

Před instalací balíků pro Python, nebo pozdějším spouštěním serverové aplikace, je potřeba aktivovat vytvořené virtuální prostředí. Aktivní virtuální prostředí je možné poznat pomocí jeho názvu, který se v terminálu objeví na začátku řádku před jménem uživatele. Deaktivace je možná použitím příkazu `deactivate`. [13]

```
source ~/eink_venv/bin/activate
```

Výpis 7: Aktivace virtuálního prostředí

4.1.8 Instalace nástroje CEF-Python

Nástroj *CEF-Python* je potřeba nainstalovat ve verzi 66.0. V současné době se jedná o nejnovější vydání. Pro funkčnost tohoto nástroje je vyžadována knihovna *Pillow*, která slouží ke zpracování obrazových dat. Správce balíků *pip* je zde spouštěn bez zadání čísla jeho verze, jelikož je volán uvnitř virtuálního prostředí. [14, 15]

```
pip install Pillow
pip install cefpython3==66.0
```

Výpis 8: Instalace knihovny Pillow a nástroje CEF-Python

4.1.9 Instalace webového frameworku Django

Poslední položkou, kterou je do virtuálního prostředí potřeba doinstalovat, je webový framework Django. Nejmenší podporovaná verze je 2.2, k dispozici jsou však i novější vydání. [16]

```
pip install django
```

Výpis 9: Instalace webového frameworku Django

V případě, že instalace probíhá na systému, ve kterém je přítomna nepodporovaná starší verze, může být provedena aktualizace na nejnovější vydání přidáním parametru `-U`.

```
pip install django -U
```

Výpis 10: Aktualizace webového frameworku Django

4.2 Klientská část

Pro klientské zařízení, kterým je modul ESP32, je potřeba pouze vývojový framework *ESP-IDF*. K jeho zprovoznění je nejprve nutné nainstalovat všechny požadované balíky.

```
sudo apt install gcc git wget make libncurses-dev flex bison gperf -y
sudo apt install python python-pip python-setuptools python-serial -y
sudo apt install python-cryptography python-future python-pyparsing -y
```

Výpis 11: Instalace požadovaných balíků pro ESP-IDF

Následuje vytvoření složky, do které je stažen a rozbalen archiv s potřebnými nástroji. [17]

```
mkdir -p ~/esp
cd ~/esp
wget https://dl.espressif.com/dl/xtensa-esp32-elf-linux64-1.22.0-80-g6c4433a-5.2.0.tar.gz
tar -xzf xtensa-esp32-elf-linux64-1.22.0-80-g6c4433a-5.2.0.tar.gz
```

Výpis 12: Stažení balíku nástrojů pro ESP-IDF

Dalším krokem je stažení *ESP-IDF*. Použitá verze je 3.3.1, jelikož na ní probíhal vývoj. Existuje i novější verze, kterou je možné také použít. V takovém případě je však potřeba přizpůsobit strukturu projektu klientské aplikace změnám, které v nové verzi nastaly. [18]

```
cd ~/esp
git clone -b v3.3.1 --recursive https://github.com/espressif/esp-idf.git
```

Výpis 13: Stažení vývojového frameworku ESP-IDF

Nyní je potřeba nastavit hodnoty dvou systémových proměnných. Do proměnné `$PATH` je přidána cesta ke složce s balíkem nástrojů. Do proměnné `$IDF_PATH` je přiřazena cesta ke složce, ve které jsou obsaženy soubory *ESP-IDF*. Aby tyto proměnné zůstaly trvale zachovány, je potřeba tyto příkazy ještě vložit do souboru `~/.profile`. [17, 19]

```
export PATH="$HOME/esp/xtensa-esp32-elf/bin:$PATH"
export IDF_PATH=~/esp/esp-idf
```

Výpis 14: Nastavení cest ke složkám ESP-IDF

Posledním krokem je instalace požadovaných balíků pro Python. [18]

```
python -m pip install --user -r $IDF_PATH/requirements.txt
```

Výpis 15: Instalace balíků pro Python, které ESP-IDF vyžaduje

5 Webová aplikace

Aplikace běžící na serveru se skládá ze dvou částí. První částí je webová aplikace, která poskytuje webové rozhraní sloužící ke konfiguraci různých položek uživatelem a k zobrazování přehledů. Také poskytuje data ke stažení, která potřebuje klientské zařízení k vykreslování obrazu na e-ink display. Tato kapitola se zabývá popisem webové aplikace. Druhou částí je vykreslovací aplikace, která je popsána v kapitole 6.

Webová aplikace používá webový framework *Django*. Jedná se nástroj založený na programovacím jazyce *Python*. Slouží především ke zrychlení a zjednodušení vývoje webu. Například poskytuje zjednodušený přístup k databázi, u kterého není potřeba psát *SQL* kód. Dále umožňuje používat *HTML* šablony, zjednodušuje práci s webovými formuláři a mnoho dalšího. [20]

5.1 Možnosti konfigurace

Webové rozhraní poskytuje možnost snadné konfigurace potřebných položek prostřednictvím webového prohlížeče. Uživatel má možnost vytvářet, aktualizovat a mazat záznamy týkající se vykreslování webových stránek nebo získávání dat rozvrhů hodin.

5.1.1 Webové stránky

Konfigurace webových stránek, které mají být vykreslovány, probíhá pomocí tří typů záznamů. Jejich názvy v následujícím výpisu odpovídají jménům databázových tabulek.

- **Display** – Tento záznam slouží k nastavení parametrů e-ink displaye připojeného k modulu ESP32. Mezi parametry patří jméno displaye, výška a šířka v pixelech, orientace obrazu a bitová hloubka jednotlivých pixelů. Orientace může být normální, otočená o 90° po směru nebo proti směru hodinových ručiček a otočená o 180°.
- **Webpage** – Hlavní položky nastavení webové stránky jsou jméno, adresa (*URL*) a interval vykreslování. Dále je možné změnit úroveň zvětšení, hranici mezi černou a bílou barvou a dvě položky pro opožděné vykreslení webové stránky. Také je k dispozici volba pro skrývání posuvníků.
- **Device** – Záznam každého klientského zařízení, které má vykreslovat webovou stránku, se skládá z výběru jednoho displaye a jedné webové stránky. Dále je potřeba zadat jméno a identifikátor. V případě potřeby je také možné nastavit klientské zařízení jako neaktivní.

5.1.2 Rozvrhy hodin

Pro konfiguraci vykreslování rozvrhů hodin vybraných místností slouží záznam typu **Schedule**. Stejně jako záznam typu **Device** obsahuje jméno, identifikátor a volitelné označení položky jako neaktivní. Místo výběru webové stránky je k dispozici výběr místnosti, jejíž rozvrh hodin má

být vykreslován. Výběr displaye zde není, jelikož na rozdíl od vykreslování webových stránek probíhá vytvoření obrazové bitmapy až na klientském zařízení.

Před zadáním místnosti je potřeba ještě zvolit budovu a areál. Pro výběr z dostupných voleb jsou použity tři rozbalovací seznamy. Jejich položky jsou pro přehlednost abecedně seřazeny. Pro rychlé vyhledávání v označeném seznamu je možné napsat pomocí klávesnice začátek názvu požadované položky.

Při vytváření nového záznamu jsou vybrány výchozí položky, které odpovídají předpokládaným nejčastějším volbám. Pro areál je vybranou položkou *Poruba*, výchozí budovou je *Fakulta elektrotechniky a informatiky*. Místnost je nastavena na jednu z přednáškových místností. Výchozí hodnoty je možné v případě potřeby změnit nebo úplně odstranit.

Výběrem některého areálu nebo budovy jsou podřízené seznamy změněny takovým způsobem, aby byly zobrazeny pouze položky, které pod danou volbu patří. Tuto funkcionalitu zajišťuje *Javascript*, který dynamicky mění obsah rozbalovacích seznamů na základě informací, které jsou asynchronně získávány ze serveru. To je prováděno pomocí *API*, které je k tomuto účelu na serveru k dispozici. Pro areály je vracen seznam všech položek, zatímco budovy a místnosti jsou filtrovány na základě zadaného identifikátoru nadřazené položky. Odpověď má jednoduchý textový formát, ve kterém je na každém řádku umístěna jedna položka. Ta se skládá z názvu, znaku vyhrazeného jako oddělovač a identifikátoru položky. Při vytváření voleb rozbalovacího seznamu je název určen pro zobrazení, zatímco identifikátor je použit při odesílání webového formuláře.

5.2 Zobrazování přehledů

Kromě nastavení obsahu k vykreslování zpřístupňuje webové rozhraní také tři přehledy. První přehled obsahuje seznam klientských zařízení a připojených e-ink displayů, které jsou použity k zobrazování nastavených webových stránek. Druhý přehled vypadá podobně, obsahuje však vykreslované rozvrhy hodin.

Oba přehledy obsahují časové informace, kdy na serveru proběhla poslední aktualizace obsahu, který má být zobrazován. Součástí jsou také informace o posledním stažení dat odpovídajícími klientskými zařízeními. Je tak možné detekovat klientská zařízení, která příliš dlouhou dobu nebyla aktualizována. To může být způsobeno například problémem v síťové komunikaci nebo přerušením napájení (například vybité baterie). Pro rychlou orientaci jsou seznamy seřazeny od nejbližšího času stažení dat po nejvzdálenější.

Třetí přehled obsahuje textový seznam pořízených snímků nastavených webových stránek, ve kterém se položky skládají z názvu webové stránky a e-ink displaye, pro který vykreslování probíhá. Po výběru některé z položek je zobrazena nová stránka, která obsahuje podrobnější informace a náhled vykresleného obrázku. Ten je k dispozici ve dvou verzích. První z nich je originální snímek webové stránky v barevném provedení. Druhý obrázek je převeden do stupňů šedi takovým způsobem, aby odpovídal zobrazení na e-ink displayi. Na obrázky je možné kliknout pro otevření v novém panelu a zobrazení v plné velikosti.

5.3 Přístup k API

Webový server poskytuje *API*, které umožňuje klientům stahovat obrazové bitmapy webových stránek nebo rozvrhy hodin vybraných místností. Server vrací data v závislosti na identifikátoru klienta, který je každému z nich přidělen. Adresu pro přístup, která musí být na klientském zařízení nastavena, je možné zkopírovat z webového rozhraní v požadovaném formátu.

Kromě souboru s daty pro vykreslení obsahu na e-ink display jsou přenášeny také další informace pomocí *HTTP* hlaviček. Jedná se velikost přenášeného obsahu kvůli kontrole množství stahovaných dat, aktuální časovou informaci, která je určena k vykreslení na display a identifikátor typu dat, pomocí kterého je možné odlišit obrazovou bitmapu od rozvrhu hodin. Pokud je typem dat obrazová bitmapa, je přenášena také informace o nastavené orientaci displeje.

5.4 Spuštění webového serveru

Spuštění webového serveru probíhá pomocí skriptu `manage.py`, kterému jsou předávány potřebné parametry. Je možné zvolit IP adresu a port, na kterém bude webový server zpřístupněn. IP adresa tvořená nulami umožňuje připojení přes libovolné síťové rozhraní. Port 8000 byl zvolen tak, aby pro jeho použití nebylo zapotřebí oprávnění uživatele *root*. Pokud je používáno virtuální prostředí (viz kapitola 4.1.5), musí být nejprve provedena jeho aktivace.

```
source ~/eink_venv/bin/activate
cd eink_server/web_app/
python manage.py runserver 0.0.0.0:8000
```

Výpis 16: Spuštění webového serveru

6 Vykreslovací aplikace

Úkolem této aplikace běžící na serveru je získávat data pro vykreslení na e-ink display a převádět je do takového formátu, který je vhodný pro klientská zařízení. Všechna data jsou připravena takovým způsobem, aby je bylo možné ihned odeslat po vyžádání klientem. V nastavených intervalech probíhá nové získávání dat, aby veškeré informace zůstávaly aktuální. Tato data se dělí na dva typy, jejichž zpracování je popsáno v této kapitole. Prvním z nich jsou webové stránky, druhým typem jsou rozvrhy hodin.

6.1 Snímky webových stránek

Aby bylo možné zobrazit webovou stránku na e-ink displayi, je potřeba nejprve zachytit její statický snímek. K tomuto účelu je použit webový prohlížeč, jehož prostřednictvím je webová stránka navštívena a vykreslena. Po vytvoření obrázku je proveden převod na požadovaný počet odstínů šedi. Následuje uložení souboru v takovém formátu, který je snadno zpracovatelný na klientském zařízení. Během tohoto procesu jsou prováděny také úpravy týkající se kvality zobrazení, které budou dále popsány. Základ tohoto postupu, včetně rozvahy nad jeho vhodností, je popsán v kapitole 2.1.

6.1.1 Chromium Embedded Framework (CEF)

Pro navštěvování, vykreslování a pořizování snímků webových stránek je použit *Chromium Embedded Framework*. Jedná se o framework s otevřeným zdrojovým kódem, který umožňuje používat webový prohlížeč *Chromium* a ovládat ho jinou aplikací. [21]

Webový prohlížeč Chromium je podobný webovému prohlížeči *Google Chrome*, funkcionality se však může mírně lišit. Chromium má totiž otevřený zdrojový kód a neobsahuje proprietární softwarové součásti, které jsou zahrnuty v Google Chrome, jako například některé multimediální kodeky. [22]

CEF umožňuje konfigurovat a ovládat webový prohlížeč Chromium pomocí poskytnutých funkcí. Jelikož jediné podporované programovací jazyky jsou *C/C++*, je místo jeho základní verze použit *CEF-Python*. Jedná se o rozšíření umožňující používání pomocí programovacího jazyka *Python*. [14]

Přestože je prováděno vykreslování webových stránek bez použití viditelného okna webového prohlížeče, současná verze pro svou činnost v Linuxu stále vyžaduje správce oken. Tohle omezení by mělo být odstraněno v některém z budoucích vydání. Pro systémy bez obrazovky může být použit software, jako je například *Xvfb*, který provádí všechny grafické operace v paměti bez zobrazování jakéhokoli obrazového výstupu. [23]

6.1.2 Plánovač vykreslování

Plánovač slouží k určení, kdy mají být webové stránky navštíveny a vykresleny, a to na základě intervalu, který je nastaven pro každou webovou stránku zvlášť. Při úspěšném zpracování webové stránky je uložena aktuální časová hodnota. Po určitou dobu je pak zabráněno tomu, aby k této webové stránce bylo znovu přistupováno.

Při každém volání plánovače, které za běhu programu probíhá v pravidelných intervalech, jsou vybrány adresy webových stránek, jejichž snímky by na základě časových údajů měly být pořízeny. Zároveň jsou určeny další parametry týkající se zpracování a uložení výsledných obrázků. Pokud dojde k naplánování alespoň jedné adresy, proběhne zapnutí webového prohlížeče a zpracování webových stránek. Poté je prohlížeč vypnut, dokud nebude znovu potřeba. Samotný *CEF*, pod který webový prohlížeč spadá, zůstává po celou dobu běhu programu inicializovaný.

Plánovač provádí optimalizace, aby bylo načítání webových stránek co nejeefektivnější. Pokud existuje vícekrát stejná adresa webové stránky, ale s rozdílným nastavením, proběhne navštívení pouze jednou. Změna rozlišení a úrovně zvětšení je prováděna u stejné načtené webové stránky. Změny je dosaženo úpravou vlastností okna webového prohlížeče, přičemž proběhne nové vykreslení obsahu. Na rozdíl od jednorázového pořízení snímku a vytváření ostatních verzí pomocí úprav jeho velikosti je zachována obrazová kvalita a správný styl webových elementů, který se může při různých velikostech stránky měnit.

Může se stát, že je naplánováno více stejně nastavených webových stránek, u kterých se liší pouze bitová hloubka jednotlivých pixelů. V takovém případě není ve webovém prohlížeči prováděna žádná další činnost. Snímek je zachycen pouze jednou a poté uložen ve více kopiích v různých počtech odstínů šedi.

Díky těmto optimalizacím je možné snížit větší množství návštěv webové stránky na jedinou. Je tak dosaženo snížení využití prostředků počítače, zmenšení zátěže sítě a zkrácení doby potřebné pro vykonání činnosti.

6.1.3 Pořizování snímků webových stránek

Poté, co je webová stránka načtena, vyvolá *CEF* odpovídající událost. Zachycením této události lze zjistit, že už je možné pořídit snímek okna webového prohlížeče a uložit ho jako obrázek. Pokud by však byl snímek pořízen okamžitě, mohlo by se stát, že by část webové stránky chyběla. Je to způsobeno tím, že vykreslování webové stránky trvá ještě malou chvíli po jejím načtení. Některé weby také zobrazují různé efekty a animace, které je potřeba nechat proběhnout, nebo používají opožděné načítání externích zdrojů. Z tohoto důvodu byla přidána možnost oddálení pořízení snímku po načtení webové stránky o daný počet sekund. Ve výchozím stavu je tato doba čekání nastavena na 2 sekundy, jelikož se tato hodnota ukázala být nejvhodnější pro většinu testovaných webových stránek.

Při návštěvě některých webových stránek je událost, která značí dokončení načítání, vyvolána více než jedenkrát. To je obvykle způsobeno načtením externí webové stránky, která je do

hlavní stránky vsazena. Typicky se jedná o reklamy, které bývají touthle technikou zobrazovány. V takových případech může být při prvním vyvolání události ještě příliš brzy na to, aby byl pořízen snímek okna webového prohlížeče. Z toho důvodu je možné nastavit počet událostí, které musí být vyvolány, aby došlo z zachycení snímku. Přestože tohle nastavení může být v některých případech užitečné, obecně je bezpečnější ho neměnit a místo toho nastavit delší zpoždění před vykreslením. Může se totiž stát, že se obsah webové stránky časem změní nebo nedojde k načtení externího zdroje, takže se změní i počet vyvolaných událostí.

6.1.4 Nastavení úrovně zvětšení

Před pořízením snímku webové stránky je možné změnit úroveň zvětšení. Nastavitelné hodnoty se pohybují v rozsahu od 25 % do 500 %, takže je možné obsah kromě zvětšení také zmenšit. Přestože *CEF* dovoluje nastavit prakticky jakoukoliv hodnotu zvětšení, uživatelé jsou zpřístupněny přednastavené hodnoty, které odpovídají volbám dostupným v běžném webovém prohlížeči.

6.1.5 Skrývání posuvníků

Obsah webových stránek je obvykle větší, než co se může vlézt na obrazovku. V takovém případě je vytvořen posuvník, který je umístěn na okraji okna. Ten v normálním případě slouží k pohybu po webové stránce. V tomto případě je však jeho vykreslení nežádoucí, protože se stává součástí webové stránky vykreslované na e-ink display.

Pro odstranění posuvníků byla přidána položka nastavení, která je ve výchozím stavu aktivní. Skrytí posuvníků probíhá pomocí vložení určitých *CSS* pravidel, která nastavují chování při přetečení obsahu. Přestože tento způsob funguje správně pro všechny testované webové stránky, může se stát, že u nějakého webu naruší jeho zobrazení. V takovém případě je možné individuální vypnutí této funkce.

6.1.6 Převod na stupně šedi

Vytvořením snímku webové stránky jsou získána data ve formě obrazové bitmapy, která je složena z jednotlivých pixelů. Každý pixel obsahuje tři hodnoty, které reprezentují jednotlivé základní barvy. Všechny tyto barevné pixely, které mají každý velikost 3 byte, je potřeba převést na pixely ve stupních šedi o velikosti 1 byte. Pro výpočet hodnoty každého nového pixelu P je použit následující převodní vzorec podle normy *ITU-R 601-2*, kde R je hodnota červené, G zelené a B modré barvy původního pixelu: [24]

$$P = R \cdot \frac{299}{1000} + G \cdot \frac{587}{1000} + B \cdot \frac{114}{1000} \quad (1)$$

Každý pixel P nyní reprezentuje jeden z 256 odstínů šedi. Nově vzniklá obrazová bitmapa je dále zpracována takovým způsobem, aby bylo dosaženo požadovaného počtu odstínů šedi. Tento počet může být buď 2, 4, 16 nebo 256. V posledním případě je však možné tento krok

přeskočit, protože by byl vytvořen stejný výstup, jako vstup. Čísla možných počtů odstínů šedi byla získána pomocí výpočtu 2^H , kde H odpovídá bitové hloubce jednotlivých pixelů a může nabývat některé z těchto hodnot: 1, 2, 4 nebo 8. Stejně velkými skupinami po tolika bitech lze totiž přesně zaplnit jeden byte.

Pro výpočet nové hodnoty pixelu s redukováným počtem odstínů šedi P_R je použit následující vzorec, který používá celočíselné dělení a bitový posun:

$$P_R = \frac{P}{1 \ll (8 - H)} \quad (2)$$

Tento vzorec odpovídá výpočtu, který by bez použití bitového posunu vypadal následovně:

$$P_R = \frac{P}{2^{8-H}} \quad (3)$$

Posledním krokem tohoto procesu je spojení více pixelů do jednoho byte. To je provedeno pomocí bitového posunu a bitové operace *OR*. Pro výpočet množství pixelů M , které je možné zároveň do jednoho byte umístit, lze použít následující jednoduchý výpočet:

$$M = \frac{8}{H} \quad (4)$$

6.1.7 Hranice černé a bílé barvy

Pro převod obrazové bitmapy z barevného formátu na stupně šedi je k dispozici také alternativní režim. Dostupný je však pouze při bitové hloubce pixelů, která má hodnotu 1, tedy při převodu na černou a bílou barvu bez ostatních stupňů šedi. Pokud jsou k dispozici další odstíny šedi, kvalita obrazu je natolik dobrá, že další úpravy nejsou potřeba.

V běžném režimu by byl každý pixel nejprve převeden na jeden z 256 odstínů šedi a poté převeden na černou nebo bílou barvu v závislosti na tom, zda by se nacházel v tmavší nebo světlejší polovině. V alternativním režimu je možné hranici mezi černou a bílou barvou nastavit manuálně. Je tak možné zabránit některým světlejším barvám, aby se staly bílou a místo toho byly zobrazeny jako černá barva. To stejné platí i obráceně.

Možnost manuálního nastavení této hranice byla přidána, protože je v některých případech obtížné automaticky převést barevný obrázek do černobílé uspokojivým způsobem. Různé webové stránky mají jiný vzhled, takže upravení této hranice pro každou stránku zvlášť obvykle přinese značné zlepšení kvality výsledného obrazu. Je tak možné například vyřešit problém s mižením textu, který je převáděn na stejnou barvu, jako jeho pozadí.

6.1.8 Zobrazení obrázku ve stupních šedi

Aby bylo možné pomocí webového rozhraní zobrazit data, která reprezentují pixely v odstínech šedi, musí být proveden převod na obrázek. Každý byte obrazových dat je potřeba rozdělit na jednotlivé pixely podle toho, jaká byla při dřívější konverzi nastavena hodnota bitové hloubky

pixelů. Rozdělení je provedeno použitím bitového posunu a bitové operace *AND*. Každý pixel je během převodu umístěn do samostatného byte. Dalším krokem je rozprostření hodnot všech pixelů po celém rozsahu 0 – 255. Toho je dosaženo vynásobením hodnotou N , která je získána pomocí následujícího vzorce:

$$N = \frac{255}{2^H - 1} \quad (5)$$

Posledním krokem je převedení nově vzniklé obrazové bitmapy na obrázek ve formátu *PNG*. K tomuto účelu je použita knihovní funkce. Pro převod je potřeba znát výšku a šířku obrázku, což jsou údaje získané z hlavičky souboru s původními obrazovými daty (viz další kapitola).

6.1.9 Ukládání souborů

Zachycené snímky webových stránek jsou ukládány jak ve formátu obrazové bitmapy ve stupních šedi, která je určena pro zobrazení na e-ink displayi, tak v originálním barevném provedení ve formátu *PNG*. Obě verze je možné společně zobrazit pomocí webového rozhraní, aby je bylo možné porovnat. Lze tak zjistit, jak webová stránka vypadala ve webovém prohlížeči a jak bude vypadat na e-ink displayi. Tohle zobrazení je užitečné také ve chvíli, kdy je potřeba zkoušet různá nastavení pro výběr toho nejvhodnějšího.

Při ukládání souboru s daty pro e-ink display jsou kromě samotného obrazu potřeba uložit ještě další informace. Ty jsou umístěny do hlavičky na začátek souboru. Hlavička se skládá ze čtyř 16-bitových čísel, za kterými následují hodnoty jednotlivých pixelů. Čísla v hlavičce jsou zakódována pomocí *little endian*, aby pořadí bytů odpovídalo způsobu uložení čísel v modulu ESP32. Význam jednotlivých položek, které jsou v hlavičce uloženy v tomto pořadí, je následující:

1. **Verze hlavičky** – Aby bylo možné v budoucnu upravit strukturu hlavičky, je potřeba mít možnost tuto změnu detekovat. Z toho důvodu udává první položka hlavičky její verzi. V případě, že dojde ke změně ve struktuře hlavičky, by měla být tato hodnota inkrementována. Klientské zařízení pak na základě této verze určí, zda je schopno následující binární data správně interpretovat.
2. **Bitová hloubka pixelů** – Informace o počtu bitů, které jsou potřeba k reprezentaci jednoho pixelu. Zároveň určuje počet pixelů, které jsou obsaženy v jednom byte. Také udává počet odstínů šedi.
3. **Šířka obrázku** – Pro správnou interpretaci obrazových dat je potřeba znát šířku obrázku. Díky ní je možné určit, kdy při vykreslování přestat vkládat pixely na současný řádek a pokračovat na začátku následujícího řádku.
4. **Výška obrázku** – Slouží především k ověření stažených dat. Je možné například zkontrolovat, zda nedošlo k chybě při konfiguraci rozměrů displaye. Dále lze společně s šířkou obrázku, bitovou hloubkou pixelů a délkou hlavičky přesně vypočítat očekávanou velikost stahovaného souboru a tuto hodnotu porovnat s množstvím stažených dat.

Když jsou soubory ukládány, je nejprve použito jméno, které je vygenerováno jako *UUID* verze 4. Jedná se o univerzální unikátní identifikátor, který je vygenerován náhodným způsobem. Po uložení souboru s tímto jménem je vytvořena atomická databázová transakce. V jejím rámci je v odpovídající tabulce vytvořen záznam s novým jménem souboru, které obsahuje identifikátory *displaye* a *webové stránky* spolu se sekvenčním číslem. Před ukončením transakce jsou přejmenovány soubory z původního na nový název. Pokud při tomto procesu cokoliv selže, je transakce místo uložení zrušena. Zůstanou tak pouze uložené soubory, které nemají žádnou vazbu na databázi. [25]

Pojmenování souborů takovým způsobem, aby neměly vazbu na databázi, je používáno i při aktualizaci obrázků webových stránek na novější verzi. Místo smazání starého souboru s obrázkem, poté co je uložen nový soubor, zůstanou oba současně existovat. Jejich název je podobný, liší se pouze v inkrementovaném sekvenčním čísle u novějšího souboru. Záznam v databázi je aktualizován takovým způsobem, aby místo starého souboru ukazoval na nový. Tímto způsobem je možné bezpečně vytvořit a začít používat nový soubor i ve chvíli, kdy je starý soubor stále ještě využíván.

6.1.10 Odstranění nepotřebných souborů

Na konci každé iterace vykreslování webových stránek je zahájeno odstraňování nepotřebných souborů. Jedná se o osamocené soubory ve složce s obrázky, na které neodkazuje žádný záznam v databázi. Při pokusu o smazání některých souborů se může stát, že v danou chvíli nemohou být smazány. V takovém případě jsou tyto soubory přeskočeny a další pokus o smazání proběhne na konci následující iterace vykreslování.

6.2 Data rozvrhů hodin

Rozvrhy hodin, na rozdíl od webových stránek, nejsou převáděny na obrazovou bitmapu. Její vytvoření probíhá až na klientském zařízení. Díky tomu je přenášeno výrazně menší množství dat a výsledný obraz je optimalizován pro konkrétní použitý e-ink display.

Data rozvrhů hodin je nejprve potřeba získat z univerzitního informačního systému pomocí rozhraní *INBUS*. Pro přístup k těmto datům musí být provedeno ověření uživatele. Stažená data jsou dále upravena takovým způsobem, aby je bylo možné snadno zpracovat na klientském zařízení. Některé doplňující informace ke zvolenému postupu jsou k dispozici v kapitole 2.2.

6.2.1 Rozhraní INBUS

Integrovaná brána univerzitních systémů je rozhraní, které slouží jako vstupní brána pro přístup k API webových služeb poskytovaných univerzitou. Funguje jako prostředník mezi uživatelem, jehož požadavky zasílané na službu jsou validovány, a konkrétní implementací služby, které je požadavek předáván.

Pro přístup ke konkrétním službám musí být zaregistrována aplikace, které jsou přiděleny požadovaná oprávnění. U ní je určen název, popis, autorizační protokol a seznam povolených služeb. Také jsou vygenerovány přihlašovací údaje, které se skládají z identifikátoru a hesla aplikace.

Pro komunikaci je k dispozici *REST API*. Dotazy probíhají pomocí jednoduchých *HTTP* požadavků na určené webové adresy. Případné parametry mohou být součástí adresy, jiné jsou zase vkládány pomocí *HTTP* hlaviček. Odpovědi jsou vráceny ve formátu *JSON*, lze však vyžádat použití formátu *XML*. K dispozici je také volba jazyka, který je možné změnit z češtiny na angličtinu.

6.2.2 Dostupné služby

Aplikace, která byla v rámci rozhraní *INBUS* pro tuto práci zaregistrována, má zpřístupněny služby pro rozvrhy hodin a také pro obecné a studijní záležitosti. Hlavní kategorie informací, ke kterým je možné přistupovat, jsou v následujícím výpisu. Také jsou zde vypsány položky, které do těchto kategorií patří.

- **Common** – Obecné záležitosti. Každá možnost vrací seznam všech dostupných položek. Také lze zadat ID pro získání konkrétní položky.
 - **educationWeeks** – Názvy týdnů, ve kterých může probíhat výuková aktivita (každý, lichý, sudý, nepravidelně).
 - **scheduleWindows** – Rozvrhová okna. Obsahují pořadí, čas začátku a čas konce.
 - **weekDays** – Dny v týdnu. Každý z nich obsahuje název v normální i zkrácené verzi.
- **Edu** – Studijní záležitosti. Obdobně jako u obecných záležitostí lze vyžádat seznam všech položek nebo pouze jednu konkrétní. Kromě použití ID je možné u některých položek zadat výběr určením data. Také je někdy možné vyžádat položku předcházející či následující.
 - **educademicYears** – Akademické roky obsahující počáteční a koncové datum.
 - **educationTypes** – Názvy typů výuky, které jsou dostupné v normální i zkrácené verzi (přednášky, cvičení, konzultace, nepřímá výuka).
 - **semesters** – Informace o jednotlivých semestrech. Obsahem je počáteční a koncové datum, počet týdnů výuky a zkoušek, typ semestru a akademický rok, do kterého semestr patří.
 - **semesterTypes** – Zkrácené a normální názvy typů semestru (zimní, letní).
 - **studyForms** – Názvy forem studia, včetně zkrácených verzí (prezenční, kombinovaná, distanční, celoživotní).
 - **studyTypes** – Celé a zkrácené názvy typů studia (bakalářské, navazující magisterské, magisterské, doktorské, univerzita 3. věku).

- **Pasport** – Místnosti, budovy a areály. Zadáním ID lze přistupovat přímo ke konkrétním položkám, jinak je vrácen celý seznam.
 - **areals** – Jednotlivé areály patřící k univerzitě. Skládají se z celého názvu a zkratky.
 - **buildings** – Názvy budov včetně zkratk. Obsahují identifikátor areálu, do kterého patří. Může zde být také informace o fyzickém umístění budov ve formě zeměpisných souřadnic. Při vytváření dotazu je možné volitelně zadat identifikátor jednoho areálu. Místo celého seznamu jsou pak vráceny pouze budovy, které jsou jeho součástí.
 - **rooms** – Informace o místnostech. Součástí je kód místnosti, který je k dispozici také v delší variantě, která navíc obsahuje zkratku areálu. Dále je zde číslo podlaží, na kterém se místnost nachází a identifikátor budovy, do které místnost patří. Mohou zde být přítomny také zeměpisné souřadnice, kapacita místnosti a informace o tom, že pro danou místnost existuje rozvrh hodin. Podobně jako u seznamu budov lze upravit dotaz tak, aby byly vyfiltrovány pouze některé položky. V tomto případě je možné zadat identifikátor budovy, pro kterou mají být místnosti vráceny.
- **Schedule** – Rozvrhy hodin. Jejich vyžádání probíhá na základě zadaného identifikátoru. Struktura vracených dat je stejná pro místnosti i pro vyučující. Obsahem je seznam výukových aktivit. Každá aktivita popisuje výuku konkrétního předmětu a skládá se z mnoha informací. Mezi nejdůležitější patří název předmětu, typ výuky, jména vyučujících, den a čas konání a délka výuky. Další užitečnou informací je seznam týdnů, ve kterých se výuková aktivita vyskytuje. Dále se zde objevují například informace ohledně místnosti, semestru a akademického roku.
 - **rooms** – Rozvrhy místností. Výběr probíhá zadáním číselného identifikátoru požadované místnosti. Volitelně je možné zadat také identifikátor semestru, pokud má být vrácen rozvrh pro jiný, než současný semestr. Také lze zadat příznak, který určuje, zda má rozvrh hodin obsahovat informace pouze pro aktuální týden.
 - **teachers** – Rozvrhy vyučujících. Výběr je určen pomocí loginu nebo číselného identifikátoru. Volitelně lze zadat počáteční a koncové datum, pokud má být rozvrh vrácen pro jiné období, než od začátku do konce aktuálního semestru.

Z dostupných informací je velká část použita, některé položky však zůstaly nevyužity. Poskytují tak možnost pro budoucí úpravu nebo přidání funkcionality, která se v tuto chvíli v aplikaci nevyskytuje.

6.2.3 Způsob ověření přístupu

Během přístupu k rozhraní *INBUS* je používán protokol *OAuth2*. Pomocí něj je zajištěna autentizace i autorizace zaregistrované aplikace, včetně uživatele, pod kterým aplikace vystupuje. Připojení využívá protokol *HTTPS*, takže je veškerá komunikace zabezpečena.

Základem je získání autorizačního tokenu. Jeho vyžádání probíhá vytvořením *HTTP* požadavku typu *POST* na danou adresu. Součástí požadavku je několik informací, které musí být zadány. Patří sem identifikátor zaregistrované aplikace, přístupové heslo a oblast služeb, pro kterou má autorizační token platit. V odpovědi je přítomen řetězec obsahující vyžádaný token, jeho typ, doba platnosti a autorizovaná oblast služeb.

Během přístupu ke službám je vždy potřeba použít získaný autorizační token. Jeho zadání probíhá vložením do *HTTP* hlavičky. Zaslání *HTTP* požadavku pak probíhá na adresu, která je dané službě přidělena. Na rozdíl od získávání autorizačního tokenu je požadavek typu *GET*. Obsah odpovědi závisí na vyžádaných datech.

6.2.4 Typy odpovědí na požadavky

Součástí odpovědí na vytvořené požadavky je kromě těla zprávy také *HTTP* stavový kód, díky kterému lze poznat, zda byl požadavek úspěšně proveden. To se týká jak získávání autorizačního tokenu, tak přístupu ke službám. Typicky je vrácen některý z následujících stavových kódů:

- **200 OK** – Vše proběhlo v pořádku a odpověď nese vyžádaná data.
- **404 Not Found** – Nebylo možné odpovědět, jelikož daná adresa neexistuje.
- **401 Unauthorized** – Zadaný požadavek nebylo možné zpracovat, protože nedošlo k úspěšné autorizaci. Tento stavový kód je vrácen při vyžádání autorizačního tokenu, pokud je špatně zadáno přístupové heslo nebo identifikátor aplikace. Pokud se tento stavový kód objeví během přístupu ke službě, znamená to, že nastal problém s autorizačním tokenem. Mohlo se stát, že jeho platnost vypršela kvůli uplynulému času nebo že bylo provedeno jeho zneplatnění. V takovém případě je potřeba vyžádat nový autorizační token.

6.2.5 Správa autorizačního tokenu

Vytvořená aplikace provádí automatickou správu autorizačního tokenu. V okamžiku, kdy je potřeba vytvořit požadavek na službu, proběhne nejprve náhled do databáze, zda je v ní potřebný token uložen. Pokud ano, je před jeho použitím ještě ověřena zbývající doba platnosti. Jestliže se v databázi použitelný autorizační token nenachází, případně pokud jeho použití vrátilo stavový kód *401 Unauthorized*, proběhne získání nového tokenu. Ten je pak uložen do databáze pro budoucí použití.

Některé operace, jako například obnovení rozvrhových informací, se skládají z většího množství po sobě jdoucích požadavků. Aby nebyla platnost autorizačního tokenu neustále znovu

kontrolována, je ověření provedeno pouze jednou před začátkem celé operace. Mohlo by se však stát, že platnost mezitím vyprší. Z toho důvodu je počítáno s platností, která je o 30 % kratší, aby bylo obnovení autorizačního tokenu vždy provedeno s dostatečným předstihem.

Jak již bylo zmíněno, pro získání autorizačního tokenu je potřeba znát identifikátor zaregistrované aplikace a přístupové heslo. Tyto údaje musí být zadány uživatelem před tím, než může být aplikace používána. V případě, že údaje při spuštění aplikace chybí, je uživatel vyzván k jejich zadání. To je potřeba provést pouze jednou, jelikož jsou uloženy do databáze pro opětovné použití. Z toho důvodu je k dispozici také možnost jejich vymazání.

6.2.6 Načtení seznamu místností

Místnost, pro kterou má být získán rozvrh hodin, je potřeba určit pomocí číselného identifikátoru. Uživateli je však poskytnuta možnost zadat místnost jejím názvem. Toho je dosaženo pomocí seznamu všech dostupných místností, které obsahují jak identifikátor, tak název. Tento seznam místností je nejprve stažen z univerzitního informačního systému a poté je vhodným způsobem zpřístupněn uživateli. Pro usnadnění orientace jsou místnosti rozděleny na skupiny podle budov, do kterých patří. Budovy také tvoří skupiny, a to na základě areálů, ve kterých jsou umístěny. Celý postup vypadá následovně:

1. Nejdříve jsou staženy seznamy obsahující všechny budovy, místnosti a areály. Stahování, včetně potřebného ověření, probíhá dříve popsáním způsobem.
2. Dalším krokem je parsování získaných dat, která jsou stažena ve formátu *JSON*. K tomuto účelu je použita funkce programovacího jazyka *Python*, která vrací data v podobě objektu. Tento objekt umožňuje snadno přistupovat k jednotlivým položkám a procházet stromovou strukturu, kterou data mohou tvořit.
3. Následuje vytvoření atomické databázové transakce. Během zapisování nových dat do databáze tak není možné mimo tuto transakci přistupovat k neúplným informacím. Přechod z jednoho konzistentního stavu do druhého je zajištěn také tím, že při úspěchu je transakce potvrzena, zatímco při výskytu jakékoliv chyby je proveden návrat do původního stavu.
4. Dále je potřeba ověřit přítomnost existujících dat, aby bylo možné informace do databáze nejen poprvé uložit, ale také je obnovit. Pokud je zjištěno, že už nějaké informace v daných databázových tabulkách existují, jsou odtud vymazány. Toho je dosaženo odstraněním všech areálů. Záznamy budov a místností jsou automaticky také odstraněny díky kaskádovému mazání.
5. Do prázdných tabulek jsou vloženy získané informace. Nejprve jsou uloženy všechny areály. Poté proběhne uložení budov, pro které jsou vytvořeny vazby cizím klíčem na odpovídající areály. Podobně proběhne následné uložení místností, pro které jsou vazby cizím klíčem tvořeny na budovy. Před ukládáním místností je kontrolováno, zda je přítomen a nastaven

příznak, který značí, že pro danou místnost je možné vypsát rozvrh hodin. Pokud tomu tak není, místnost uložena nebude.

6. Uložená data jsou zkontrolována za účelem nalezení a odstranění nepotřebných záznamů. Nejprve jsou vymazány záznamy budov, na které nemá vazbu ani jedna místnost. Tyto budovy jsou obvykle prázdné z toho důvodu, že všechny jejich místnosti byly odstraněny kvůli absenci rozvrhu hodin. Následně jsou obdobným způsobem smazány areály, které zůstaly prázdné. Na některé z nich totiž nemusí existovat žádná vazba kvůli budovám, které byly odstraněny.
7. Posledním krokem je ukončení databázové transakce. Pokud vše proběhlo v pořádku, databáze nyní obsahuje aktuální odfiltrované informace.

Tyto informace jsou použity v rámci webového rozhraní pro vytváření nebo úpravu záznamů, na základě kterých jsou získávány a vykreslovány rozvrhy hodin pro nastavené místnosti. Z toho důvodu by mělo načtení těchto informací proběhnout dříve, než je spuštěn webový server. K tomu slouží jeden z parametrů pro spuštění aplikace.

6.2.7 Získání rozvrhových dat

Data rozvrhů hodin je potřeba stáhnout, vybrat z nich potřebné informace a na jejich základě vytvořit nový soubor ve formátu, který je vhodný pro klientská zařízení. Jednotlivé kroky tohoto procesu vypadají následovně:

1. Nejprve jsou stažena rozvrhová data. Následuje převod z formátu *JSON*. Tento postup je obdobný jako při získávání seznamu místností.
2. Ze získaných dat jsou vybrány pouze položky, které budou dále potřeba. Pro každou rozvrhovou aktivitu je vytvořena jedna instance třídy, která slouží pro uchování relevantních informací. Součástí jsou také funkce, které vracejí data ve vhodnější podobě. Výsledkem je list těchto objektů.
3. Dalším krokem je odstranění diakritiky. Toho je dosaženo pomocí *Unicode* dekompozice znaků. Každé písmeno obsahující diakritiku je rozděleno na základní písmeno a diakritické znaménko. Například písmeno „č“ je nahrazeno za „c“ a „ˇ“. Následně jsou vybrány pouze základní znaky a ostatní jsou odstraněny.
4. Původní textová data jsou uložena pomocí kódování *UTF-8*. Jejich zpracování by však na klientském zařízení bylo problematické, proto je proveden převod na kódování *ASCII*. Zůstanou tak pouze znaky, které má klient možnost vykreslit pomocí rastrového písma, které k vykreslování textu používá. Výhodou je také pevná velikost jednoho znaku, která je rovna jednomu byte.

5. Z připravených informací o jednotlivých aktivitách je vytvořen soubor obsahující všechna potřebná data. Popis formátu tohoto souboru je v kapitole 6.2.8.
6. Posledním krokem je uložení výsledného souboru. Jelikož se jedná o malé množství dat, umístění probíhá přímo do databáze. Součástí záznamu je také aktuální časová informace, díky které je později možné posoudit aktuálnost rozvrhových dat.

Rozvrhová data jsou získávána ve chvíli, kdy jejich stáří překročí nastavenou hodnotu nebo v případě, že jejich první stažení ještě vůbec neproběhlo. Vybírání rozvrhů hodin, které mají být staženy a uloženy, se děje v rámci kontroly, která probíhá v pravidelných intervalech. Všechny vybrané rozvrhy jsou po kontrole zpracovány najednou.

Kontrola se opakuje i v době, kdy by podle časových informací, které jsou uloženy v databázi, nemělo být delší dobu vyžadováno žádné stahování rozvrhových dat. Může se totiž stát, že uživatel mezitím vloží záznam pro nový rozvrh hodin nebo upraví existující záznam takovým způsobem, aby obsahoval jinou místnost.

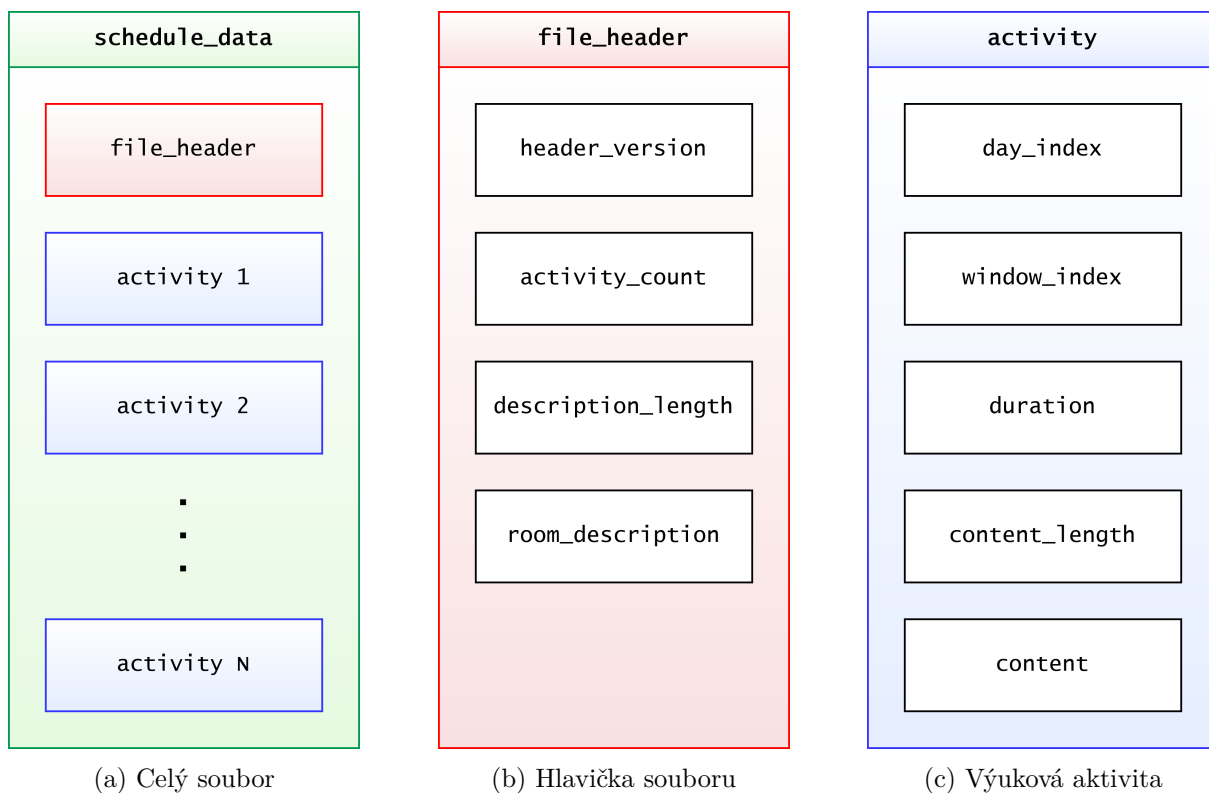
Pokud během zpracovávání dávky rozvrhů hodin dojde k chybě, je probíhající činnost přerušena. Je tak zabráněno dalšímu opakování stejné chyby, kterou je typicky problém v síťové komunikaci. Při další kontrole jsou pak znovu vybrány rozvrhy hodin, které kvůli chybě ještě nebyly zpracovány.

6.2.8 Formát souboru s rozvrhovými daty

Rozvrhová data jsou ukládána do souboru, jehož formát je stavěn tak, aby byl na klientském zařízení umožněn snadný přístup k informacím pomocí struktur programovacího jazyka *C*. Grafické znázornění formátu souboru je k dispozici na obrázku 4, který se skládá ze tří částí.

První část je obrázek 4a, který zobrazuje celý soubor. Na jeho začátku se nachází hlavička, která je vždy v souboru přítomna. Za ní následují jednotlivé výukové aktivity. Těch se zde může nacházet libovolné množství. V případě prázdného rozvrhu hodin není přítomna žádná aktivita a obsahem souboru je pouze hlavička. Ta je znázorněna na obrázku 4b a skládá se z následujících čtyř položek:

1. **header_version** – Verze hlavičky. Má stejný význam, jako u souborů obsahující snímky webových stránek (viz kapitola 6.1.9).
2. **activity_count** – Počet aktivit, které následují za hlavičkou.
3. **description_length** – Délka následujícího textu s popisem místnosti. Součástí délky není nulový znak, který textový řetězec ukončuje.
4. **room_description** – Popisek místnosti obsahující text ke zobrazení na e-ink displayi nad tabulkou s rozvrhem hodin. Jeho obsahem je název místnosti, v případě potřeby je však možná změna na libovolnou textovou informaci.



Obrázek 4: Struktura souboru s rozvrhem hodin

Na obrázku 4c je znázorněna struktura výukové aktivity. Ta se skládá z následujících pěti částí:

1. **day_index** – Index dne, ve kterém se výuková aktivita vyskytuje. Odpovídá číslu řádku tabulky s rozvrhem hodin, do kterého má být aktivita umístěna.
2. **window_index** – Index rozvrhového okna, ve kterém výuková aktivita začíná. Určuje číslo sloupce tabulky a udává čas začátku výuky.
3. **duration** – Trvání aktivity, které je zadáno počtem rozvrhových oken.
4. **content_length** – Délka následujícího textového obsahu výukové aktivity. Nezahrnuje ukončující nulový znak.
5. **content** – Textový řetězec obsahující typ výuky, název předmětu a jména vyučujících.

6.3 Spuštění vykreslovací aplikace

Aplikace by měla běžet vždy, když je spuštěn webový server. Je tak možné stále udržovat všechna data v aktuálním stavu. Prvním krokem je aktivace virtuálního prostředí, pokud je používáno (viz kapitola 4.1.5). Následuje přesun do složky s aplikací a její spuštění.

```
source ~/eink_venv/bin/activate
cd eink_server/draw_app/
python main.py
```

Výpis 17: Spuštění vykreslovací aplikace

Při spouštění aplikace je potřeba zadat některý z parametrů příkazové řádky v závislosti na činnosti, která má být vykonána. Pokud žádný parametr není použit, nebo je zadán špatně, proběhne vypsaní nápovědy. Každý z dostupných parametrů má kratší a delší možnost zápisu. U některých parametrů je potřeba zadat také další hodnoty. Dostupné možnosti jsou následující:

- **-h, --help** – Vypíše stejnou nápovědu, která je zobrazena v případě, že není zadán žádný odpovídající parametr. Obsahem nápovědy jsou všechny dostupné možnosti parametrů příkazové řádky.
- **-s, --start** – Zahájí hlavní činnost aplikace, která zůstává běžet tak dlouho, dokud není uživatelem vypnuta. Ukončení probíhá stiskem kláves **CTRL-C**. Před prvním spuštěním je potřeba zadat některé informace pomocí odpovídajících parametrů. Pokud je aplikace spuštěna bez nich, proběhne vypsaní chybějících informací a aplikace se ukončí.
- **-i, --info** – Zobrazí vybrané informace, které jsou uloženy v databázi. Jedná se hodnoty, které lze nastavit odpovídajícími parametry. U těchto hodnot je zobrazován jejich název a obsah, s výjimkou přihlašovacích údajů. U nich je zobrazena pouze informace o tom, zda se v databázi nacházejí nebo ne. Je tak možné například zkontrolovat, zda skutečně došlo k jejich smazání při použití parametru **-f**.
- **-e, --enter-credentials** – Slouží k zadání přihlašovacích údajů, které jsou vyžadovány pro přístup k rozvrhovým datům pomocí rozhraní *INBUS*. Jelikož jsou tyto údaje opakovaně potřeba pro získávání autorizačního tokenu, jsou v databázi uloženy v běžné textové podobě. Za tímto parametrem musí být zadány dvě hodnoty:
 - **app_id** – Identifikátor aplikace.
 - **app_password** – Heslo aplikace.
- **-f, --forget-credentials** – Provede vymazání přihlašovacích údajů z databáze. Pokud je zde uložen i autorizační token, je zároveň také vymazán.

- **-u, --urls** – Změní dvě webové adresy, pomocí kterých je přistupováno k rozhraní *INBUS*. Při vypsání nápovědy jsou zobrazeny jejich výchozí hodnoty. Nové webové adresy musejí být zadány tak, aby jejich formát odpovídal formátu, ve kterém jsou výchozí adresy. Za tímto parametrem následují dvě hodnoty:
 - **oauth_token** – Webová adresa pro získání autorizačního tokenu.
 - **service** – Webová adresa pro přístup k požadované službě.
- **-m, --max-schedule-age** – Nastaví životnost rozvrhů hodin. Jedná se o dobu, která určuje, za jak dlouho po stažení rozvrhových dat by mělo dojít k novému stažení aktuální verze. Zde je zadávána jedna hodnota:
 - **minutes** – Doba životnosti v minutách.
- **-l, --load-schedule-data** – Provede stažení rozvrhových dat, která jsou potřeba před zahájením další činnosti. Jedná se o seznam areálů, budov a místností, pro které existují rozvrhy hodin. Při opětovném spuštění jsou staré informace nahrazeny novými.
- **-d, --delete-schedule-data** – Vymaže z databáze rozvrhová data, která byla získána použitím parametru **-l**.
- **-r, --reset-settings** – Obnoví nastavení na výchozí hodnoty.

7 Klientská aplikace

Aplikace běžící na klientském zařízení má za úkol stahovat v pravidelném intervalu data ze serveru a na jejich základě vykreslovat nový obsah na připojený e-ink display. Tímto zařízením je vývojová deska ESP32 (viz kapitola 3.2), obecně by však bylo možné použít i jiné zařízení.

Obsahem této kapitoly je popis činnosti klientské aplikace. Některé informace se týkají pouze vykreslování webových stránek, jiné zase zpracování rozvrhů hodin. Těmto informacím předchází popis toho, co je pro obě části společné.

7.1 Společná část

Vykreslování webových stránek a rozvrhů hodin se liší ve zpracování stažených dat, ostatní fáze činnosti zůstávají stejné. V této podkapitole je nejprve stručně popsán průběh jedné iterace vykreslování pomocí klientské aplikace. Následují doplňující informace, které se týkají stahování dat pomocí WiFi, formátu obrazových dat, reagování na chyby a vypisování informací.

7.1.1 Průběh jedné iterace

Když je modul ESP32 poprvé připojen ke zdroji napájení, nebo když je probuzen z režimu hibernace, je ihned zahájena jeho činnost. Nejprve proběhne pokus o připojení k síti WiFi. V případě úspěšného připojení jsou staženy základní informace, které jsou zkontrolovány. Pokud je vše v pořádku, jsou pomocí stejného připojení stažena data pro zobrazení obsahu na e-ink displayi. Způsob jejich následného zpracování a vykreslení záleží na tom, zda se jedná o obrazovou bitmapu nebo data rozvrhu hodin. Nakonec modul ESP32 přejde do režimu hibernace, ve kterém vyčká nakonfigurovanou dobu do zahájení další iterace. Po uplynulém čase dojde k probuzení a celý proces je zopakován.

7.1.2 Připojení k WiFi

Modul ESP32 podporuje různé typy zabezpečení sítě WiFi. Pro každý typ zabezpečení je potřeba provést odpovídající konfiguraci. V tomto případě je pro připojení použit bezpečnostní protokol *WPA2-PSK*, u kterého je potřeba zadat pouze název sítě WiFi a heslo.

Pro maximální délku pokusu o připojení k síti WiFi je nastaven časový limit, který je možné v případě potřeby upravit. Pomocí něj je zamezeno příliš dlouhému snažení se o připojení k nedosažitelné síti a zbytečnému vybíjení baterií.

7.1.3 Stažení a ověření dat

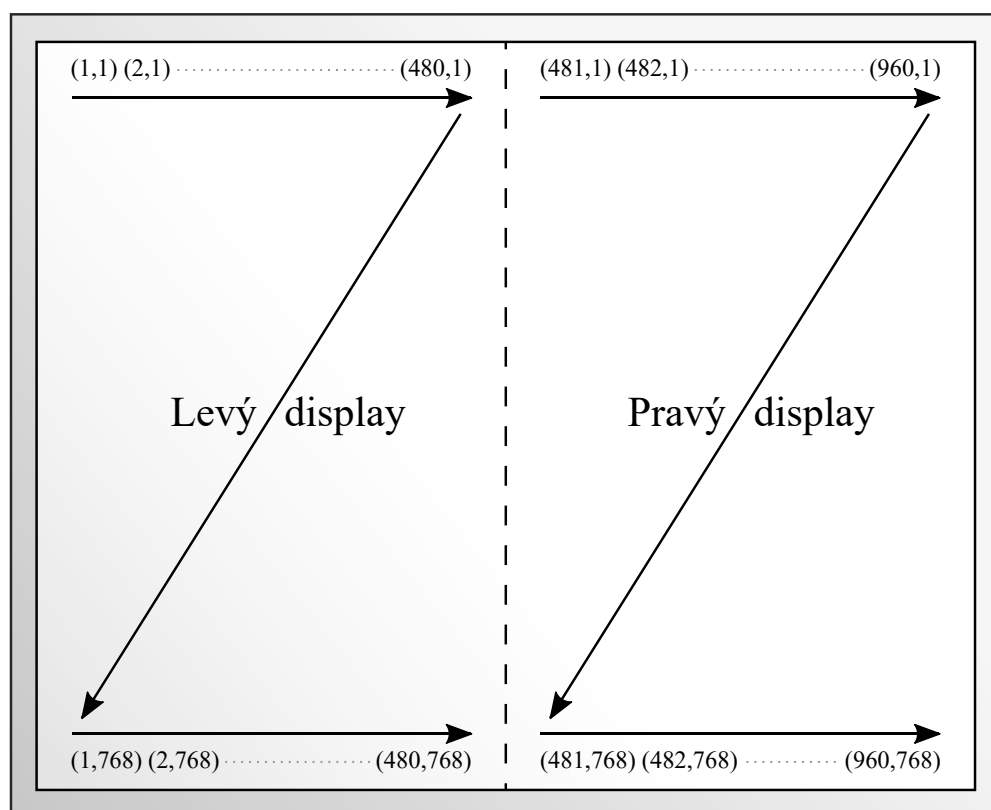
Když je zahájeno stahování dat ze serveru, nejprve jsou staženy *HTTP* hlavičky (viz kapitola 5.3). Jedna z hlaviček udává celkovou velikost následujícího těla zprávy. Po stažení rozvrhu hodin je díky ní možné ověřit, zda bylo přijato správné množství dat. Obrazová bitmapa má pevnou velikost, takže případný nesouhlas velikosti lze detekovat ještě před jejím stažením.

Další důležitá hlavička určuje typ dat, na základě kterého je se stahovanými daty zacházeno jako s obrazovou bitmapou nebo rozvrhem hodin. Pokud by tato hlavička nebyla správně nastavena, vykreslení obsahu neproběhne. Ostatní hlavičky obsahují doplňující informace a nejsou striktně vyžadovány. Kromě hlaviček je kontrolován také stavový kód odpovědi.

7.1.4 Formát obrazových dat

Obrazová data jsou nejprve uložena, aby mohla být později předána e-ink displayi k vykreslení. Jelikož je display fyzicky rozdělen na dva displace o poloviční velikosti, které jsou umístěny vedle sebe, musí být obrazová data odpovídajícím způsobem nachystána. Každý řádek obrazu je potřeba rozdělit na dvě poloviny, uložit první polovinu do jednoho datového pole, které bude zpracováno levým displayem a zbytek řádku uložit do druhého datového pole pro pravý display. Tato dvojice datových polí musí existovat zvlášť pro černou a červenou barvu, takže celý výsledný obraz je tvořen celkem čtyřmi datovými poli.

Pro reprezentaci obrazového pixelu jedné barvy je použit jeden bit, kde 0 odpovídá bílé barvě a 1 černé nebo červené barvě. V prvním byte datového pole je uložen první až osmý pixel prvního řádku, další byte obsahuje devátý až šestnáctý pixel a tak dále. Na obrázku 5 je znázorněno rozložení e-ink displaye, kde je označeno umístění pixelů číslem sloupce a řádku. [26]



Obrázek 5: Rozložení e-ink displaye [26]

7.1.5 Reakce na chyby

Pokud za běhu programu dojde k závažné chybě, není možné ve vykonávání činnosti dále pokračovat. V takovém případě je aktivován režim hibernace a činnost je opakována po pozdějším probuzení. Existují také méně závažné chyby, při kterých není potřeba vykonávání programu zastavit. Některé z nich je možné ignorovat, jiné se dají do jisté míry vyřešit. Příkladem drobné chyby může být například problém s vykreslením textu s dodatečnými informacemi nebo chybné nastavení otočení obrazu.

Mezi závažné chyby lze například zařadit problémy s připojením k síti WiFi nebo stažení nesprávného množství dat. Dále se mohou objevit nesrovnalosti ve stahovaných informacích, které jsou porovnávány s očekávanými hodnotami. Jedná se například o šířku či výšku obrazu v pixelech, bitovou hloubku pixelů nebo typ souboru. Tento druh problému je typicky způsoben chybou v konfiguraci.

7.1.6 Vypisované informace

Informace o běhu programu mohou být průběžně vypisovány do terminálu, pokud je modul ESP32 během své činnosti připojen k počítači. Každá informace je z důvodu přehlednosti zařazena do jedné z pěti úrovní výpisu. Tyto úrovně jsou odstupňovány podle jejich důležitosti. Nejdůležitější informace jsou hlášení o chybách, následují varování, běžné informace a dvě úrovně ladících informací, které slouží především pro vývoj aplikace. V závislosti na tom, o jakou úroveň důležitosti se jedná, jsou vypisované informace zobrazeny v terminálu pomocí odpovídajících barev. Jednotlivé zprávy jsou také opatřeny písmenem značícím název úrovně, které je vypsáno na začátku řádku.

Pro zaznamenávání zpráv je dostupných několik maker, jedno pro každou úroveň výpisu. Prvním parametrem je textová informace, která identifikuje aktuální soubor se zdrojovým kódem. Následuje řetězec s libovolným textem. Do něj je možné vkládat formátovací značky obdobným způsobem, jako u funkce `printf`. Při jejich použití musí následovat odpovídající počet dalších parametrů.

Mnohdy není potřeba, aby byly vypisovány veškeré dostupné informace. V takovém případě je možné omezit výpis pomocí nastavení jedné z úrovní. Žádná informace nižší důležitosti pak není vypisována. Jeden ze způsobů nastavení zajistí, aby při běhu programu bylo rozhodováno, zda informaci vypsát nebo přeskočit. Vhodnějším způsobem nastavení je druhá možnost, u které jsou nepotřebné části odstraněny už během kompilace kódu a nejsou tak vůbec součástí výsledného programu. [27]

7.2 Vykreslování webových stránek

Soubor obsahující obrazová data snímku webové stránky není potřeba téměř nijak dále zpracovávat. Data jsou totiž připravena takovým způsobem, že je lze přímo předat e-ink displayi k vykreslení. Jedinou operací nad obrazovými daty je ukládání do polí během jejich stahování. Způsob ukládání se liší v závislosti na orientaci obrazu. Před vykreslením snímku webové stránky na e-ink display je do rohu obrazu ještě doplněn text, který obsahuje časovou informaci o poslední provedené aktualizaci.

7.2.1 Otáčení obrazu

Kromě vykreslení obsahu na display klasickým způsobem je možné provést otočení obrazu po krocích odpovídajících 90° rotaci. Způsob otočení je určen v *HTTP* hlavičce získané při stahování dat ze serveru a musí být v souladu s rozměry vykreslovaného obrazu. Při rotaci o 90° po nebo proti směru hodinových ručiček jsou totiž hodnoty výšky a šířky obrazu přehozeny.

Pro vykreslení obrazu otočeného o 180° je potřeba začít plnit datová pole s obrazovými daty od konce a pokračovat směrem k začátku. Také je potřeba obrátit pořadí bitů v každém byte, protože hodnota každého bitu odpovídá jednomu obrazovému pixelu. Pro urychlení obracení pořadí bitů je předem připraveno datové pole s 256 položkami, kde každá hodnota odpovídá svému indexu, avšak s obráceným pořadím bitů.

Další dvě rotace přehazují řádky a sloupce obrazu. K tomu je potřeba rozdělit byte, který reprezentuje 8 pixelů na řádku, na jednotlivé bity. Každý z těchto bitů je následně uložen na okraj nového samostatného byte, čímž z řádku pixelů vznikne sloupec. Obsah jednoho byte je tak rozprostřen do 8 byte, přičemž u každého byte je využit pouze jeden bit. Pro využití zbytku je potřeba tuto operaci zopakovat ještě sedmkrát, s tím rozdílem, že pozice pro uložení bitu je postupně posouvána. Opakováním tohoto postupu pro všechna data je otočen celý obraz.

7.2.2 Text s informacemi

Aby bylo při pohledu na e-ink display možné určit, kdy byl zobrazovaný obsah naposledy aktualizován, je tato informace vykreslována zároveň se snímkem webové stránky. Datum a čas posledního stažení dat je součástí *HTTP* hlavičky. Tato informace je přenášena jako textový řetězec, který je možné přímo použít k vykreslení na display, jelikož formátování je prováděno už na serveru.

Obsah textového řetězce může být volitelně změněn například na jiný formát času nebo data, případně na úplně jinou informaci. Tento text je vykreslován v pravém spodním rohu e-ink displaye. Při vykreslování je brán ohled na nastavenou orientaci obrazu, aby byl text přítomen na správném místě i při otočeném zobrazení.

Pro vykreslení tohoto textu bylo zvoleno rastrové písmo s pevnou velikostí všech znaků, která je 8 pixelů na výšku i na šířku. Tato velikost byla vybrána, protože text pak na displayi nezapere příliš mnoho místa. Písmo však není příliš malé na to, aby byl text špatně čitelný. Výhodou je

i stejná velikost všech znaků, díky které je možné snadno vypočítat umístění na displayi, na které mají být vykreslovány jednotlivé znaky textového řetězce.

Vykreslení textu probíhá pomocí překrytí přes existující obraz. Samotné znaky jsou vykreslovány černě a jejich pozadí bíle. Výsledný text by však někdy mohlo být obtížné přečíst kvůli splývání jeho okrajů se sousedícím obrazem. Z toho důvodu byla přidána bílá linka, kterou je plocha s textem obtažena, aby došlo k oddělení obsahu a zlepšení čitelnosti.

7.3 Zpracování rozvrhů hodin

Na základě staženého souboru s rozvrhem hodin je potřeba vytvořit obrazovou bitmapu, která bude předána e-ink displayi k vykreslení. Rozvrh hodin je prezentován ve formě tabulky, která obsahuje informace pro aktuální týden. V jiných týdnech může rozvrh hodin vypadat jinak, především kvůli nepravidelné výuce studentů kombinovaného studia. Kvůli relativně malému prostoru, který je na e-ink displayi dostupný k vykreslení celé rozvrhové tabulky, je použito několik technik, které se s nedostatkem místa pomáhají vypořádat. Nad tabulkou je umístěn název místnosti, pro kterou rozvrh hodin platí, a časová informace o poslední provedené aktualizaci.

7.3.1 Postup vykreslení obrazu

Proces vytvoření obrazové bitmapy z rozvrhových dat se skládá z řady jednotlivých kroků. Jejich pořadí je dáno tím, že pro vykonání každého kroku je vyžadováno dokončení jednoho nebo více předchozích kroků. Výjimkou je první krok, pro který musí pouze existovat datová pole pro uložení obrazových dat. V následujícím postupu jsou tyto kroky popsány.

1. Nastavení obsahu polí s obrazovými daty na samé nuly. Vznikne tak bílá plocha, na kterou bude probíhat následné vykreslování.
2. Namapování struktur programovacího jazyka *C* na stažená data. Formát souboru, který je těmito daty tvořen, je popsán v kapitole 6.2.8. Nejprve vznikne jeden ukazatel na hlavičku souboru, která se nachází na jeho začátku. Následuje vytvoření datového pole s ukazateli na výukové aktivity, jejichž počet je uveden v hlavičce souboru. Umístění každé aktivity v souboru je určeno na základě umístění a délky předchozí položky. Délka se skládá z pevné velikosti struktury a délky následujícího textového řetězce.
3. Pro úsporu místa jsou v tabulce vykreslovány pouze sloupce, které obsahují alespoň jednu výukovou aktivitu. Pro označení, které sloupce jsou nebo nejsou prázdné, je použito datové pole typu `bool` s počtem položek odpovídajícím počtu sloupců. V tomto kroku jsou používány informace o výukových aktivitách, které obsahují index sloupce, ve kterém aktivita začíná a délku trvání, která určuje počet následujících sloupců.
4. Podobným způsobem, jako jsou zaznamenávány prázdné sloupce k vymazání, jsou určovány prázdné řádky. Rozdíl je pouze v tom, že z informací o výukových aktivitách je

použit index řádku, na kterém se aktivita nachází. Všechny prázdné řádky však odstraňovány nejsou, jelikož by jejich absence zhoršila orientaci v tabulce. Také zabírají pouze malé množství místa, kterého je nedostatek spíše na šířku než na výšku. Místo toho je k dispozici možnost odstranění určeného množství řádků z konce tabulky. To je vhodné zejména, když v tabulce nemá být zobrazován víkend, pokud výuka probíhá pouze od pondělí do pátku. Odstraňování řádků probíhá postupně od konce týdne a zastaví se, pokud narazí na řádek obsahující výuku. Pokud by se například objevila výuka v sobotu, tabulka bude obsahovat šest řádků místo pěti.

5. Výpočet velikosti odsazení, které je používáno uvnitř buněk tabulky. Určuje počet pixelů mezery, která je vkládána mezi okraje buňky a text, který je uvnitř obsažen. Větší odsazení zlepšuje přehlednost, zabírá však více místa. Proto je jeho velikost počítána dynamicky na základě místa, které je dostupné. Nejmenší a největší možná hodnota odsazení je manuálně nastavena. Použitá hodnota, která se nachází v tomto rozsahu, je vypočítána na základě poměru počtu využitých a prázdných sloupců.
6. Názvy dnů v týdnu, které tvoří první sloupec tabulky, je možné zapsat celým názvem nebo zkratkou o délce dvou znaků. Pokud je počet využitých sloupců tabulky větší než polovina, jsou pro úsporu místa použity zkratky. V opačném případě jsou zachovány celé názvy.
7. Jelikož kvůli úspoře místa nejsou vykreslovány sloupce, které jsou prázdné, v tabulce se někdy vedle sebe objeví výukové aktivity, mezi kterými je ve skutečnosti jedno nebo více prázdných rozvrhových oken. Přestože to lze poznat z časů výuky v hlavičkách sloupců, chybějící sloupce nejsou na první pohled patrné. Z toho důvodu je mezi každými dvěma sousedícími sloupci, které na sebe nenavazují, vykreslen úzký prázdný sloupec. Ten je vždy pro každou dvojici jeden, nehledě na to, kolik sloupců mezi nimi bylo vymazáno.
8. Výpočet šířky plochy, která je dostupná pro vykreslování rozvrhových oken. Nejprve je sečtena šířka prvního sloupce tabulky obsahujícího názvy dnů v týdnu, šířky prázdných oddělovacích sloupců a velikost linek na okrajích tabulky. Poté je tohle číslo odečteno od celé šířky displaye.
9. Výpočet šířky sloupců tabulky. Ta je určena stejně pro všechny sloupce s výukovými aktivitami, a to takovým způsobem, aby bylo využito co nejvíce z šířky dostupného místa. Výjimkou je situace kdy je v tabulce pouze malé množství sloupců. Pro takové případy je nastaven limit šířky, aby bylo zabráněno příliš velkému roztahení do stran. Od vypočítané šířky sloupců je odečtena velikost odsazení a tloušťka čáry tvořící okraj každé buňky, aby zbyla pouze šířka určená pro obsah buněk. To je vyžadováno kvůli pozdějšímu vykreslení textového obsahu.
10. Časy výuky jsou umístěny v prvním řádku tabulky. Skládají se z času začátku, pomlčky a času konce výuky. Mohou být vykresleny v jednom ze dvou formátů. Pokud je v tabulce

malé množství sloupců, jejich šířka je dostatečná na to, aby byly časy výuky vykresleny na jeden řádek. Pokud není ve sloupcích dostatek místa, proběhne odstranění pomlčky z časů výuky a rozdělení na dva řádky.

11. Pro každou výukovou aktivitu je potřeba zvlášť vypočítat šířku, která je v tabulce dostupná pro vykreslení jejího obsahu, jelikož se aktivita může skládat z více rozvrhových oken. Nejprve je vynásobena šířka sloupce a počet rozvrhových oken, ze kterých se výuková aktivita skládá. Poté je za každý přesah do dalšího sloupce přičtena šířka čáry oddělující buňky a odsazení okolo této čáry.
12. Kvůli omezené šířce plochy pro vykreslování textového obsahu výukových aktivit je potřeba provést rozdělení textu na jednotlivé řádky. Za tímto účelem byl vytvořen algoritmus, který je popsán v kapitole 7.3.2. Zároveň je provedeno nalezení místa v textu, kde se nachází znak vyhrazený jako oddělovač názvu předmětu a jmen vyučujících. Ten je z textu odstraněn, aktuální řádek je ukončen a index nového řádku je uložen pro pozdější vykreslení tohoto textu.
13. Na rozdíl od sloupců, které mají všechny stejnou šířku, má každý řádek určenou svoji vlastní výšku. Základem je výška textu v první buňce každého řádku, která obsahuje název dne v týdnu. Průchodem přes výukové aktivity na stejném řádku je výška zvětšována, aby nakonec odpovídala nejvyšší buňce. Výpočet výšky řádku zahrnuje velikost odsazení od okrajů buňky a výšku všech řádků textu. Mezi jednotlivými řádky je vkládána ještě malá mezera, aby sousedící řádky navzájem nesplývaly.
14. Vytvoření nadpisu pro vykreslení nad tabulkou s rozvrhem hodin. Jeho obsahem je popis místnosti a aktuální časová informace, aby bylo možné poznat, kdy proběhla poslední aktualizace. Pokud je text příliš dlouhý, proběhne jeho zalomení na více řádků.
15. Tabulka s rozvrhem hodin je vykreslována na střed e-ink displaye. Toho je dosaženo připočtením hodnoty posunu k souřadnicím všech elementů, ze kterých se tabulka skládá. Posun na ose X je vypočítán jako polovina šířky displaye, která není využívána tabulkou. Pro osu Y je výpočet stejný, pouze s výškou místo šířky. Výška a šířka tabulky jsou vypočítány jako součet rozměrů všech elementů, ze kterých se tabulka skládá. Během určování hodnoty posunu na ose Y je zajištěno, aby nad tabulkou vždy zůstalo dostatek místa pro text s nadpisem.
16. Vykreslení připraveného nadpisu nad tabulku. Pro tento text je použito větší písmo, než pro text uvnitř tabulky. Pokud je na displayi dostatek prostoru, je mezi tabulkou a nadpisem vytvořena větší mezera pro zlepšení přehlednosti. V případě vyšších tabulek je tato mezera podle potřeby zmenšována.
17. Vypočítání umístění, na kterých se nachází jednotlivé řádky tabulky. Poloha prvního řádku je dána hodnotou posunutí tabulky. Pro zjištění umístění každého následujícího řádku je

sečtena výška předchozího řádku, velikost linky, která řádky odděluje a souřadnice Y , na které se předchozí řádek nachází.

18. Podobným způsobem, jakým jsou určována umístění řádků, jsou vypočteny hodnoty souřadnic X , na kterých se nachází jednotlivé sloupce.
19. Vykreslení jmen dnů v týdnu do prvního sloupce tabulky. Zároveň jsou vykresleny všechny horizontální linky, kterými je tabulka tvořena. Tyto dvě operace probíhají současně, protože jsou pro ně shodné počáteční souřadnice. Linky tvořící okraje tabulky a spodní okraj prvního řádku jsou vykreslovány tlustší čarou. Ta je tvořena více linkami o šířce 1 px, které jsou kresleny vedle sebe.
20. Do prvního řádku tabulky jsou vykresleny časy výuky. Také jsou vykresleny vertikální linky, včetně tří tlustších. Postup je podobný, jako v předchozím bodu. Během vykreslování jsou přeskakovány sloupce, které jsou prázdné. Jsou však vykreslovány úzké oddělovací sloupce, které neobsahují ani čas výuky.
21. Pokud výukové aktivity zabírají více rozvrhových oken, je potřeba na určitých místech v tabulce propojit po sobě jdoucí buňky. Toho je dosaženo vykreslením krátkých bílých linek na odpovídajících místech, aby překryly původní černé linky.
22. Vykreslení textového obsahu všech výukových aktivit. Každá aktivita se skládá z jednotlivých řádků textu, které jsou postupně umísťovány na odpovídající souřadnice. Na začátku textu je obsažen typ výuky, který je vykreslen černou barvou. Po nalezení oddělovače, který je umístěn mezi typem výuky a názvem předmětu, je barva textu změněna na červenou. Na začátku řádku, který obsahuje jména vyučujících, je barva opět změněna na černou. Tento řádek je určen indexem, který je nalezen v jednom z předchozích kroků.
23. Posledním krokem tohoto procesu je uvolnění paměti, která byla postupně dynamicky alokována. Pokud všechny kroky proběhly v pořádku, je nyní celý obraz dokončen a připraven k zobrazení pomocí e-ink displaye.

Z tohoto postupu byly vynechány méně důležité informace. Jedná se například o způsob vykreslování znaků, linek a jednotlivých pixelů, jelikož jsou to převážně výpočty souřadnic a indexů nebo bitové operace.

Pokud není v rozvrhových datech žádná výuková aktivita, jsou všechny sloupce místo odstranění ponechány. Stejně tak neproběhne případné odstranění řádků. Buňky tabulky jsou zvětšeny, protože by bez textového obsahu byly příliš úzké. Je tak vykreslena prázdná tabulka, která značí, že pro aktuální týden není pro místnost vypsána žádná výuka.

Při vykreslování obsahu buněk tabulky jsou textové informace zarovnávány na střed v závislosti na tom, o jaký typ buňky se jedná. Buňky v prvním řádku tabulky, které obsahují časy výuky, jsou zarovnávány horizontálně. Všechny ostatní buňky jsou zarovnávány vertikálně.

Pro text uvnitř tabulky bylo zvoleno rastrové písmo, které má výšku i šířku 8 px. Jedná se o stejné písmo, které je použito k vykreslení času poslední aktualizace při zobrazování webových stránek. Pro text nadpisu bylo zvoleno větší písmo, které má výšku 16 px a šířku 12 px.

7.3.2 Zalamování řádků

Některé textové informace jsou příliš dlouhé na to, aby se vlezly na jeden řádek. Z toho důvodu byl vytvořen algoritmus, který rozděluje vstupní text na jednotlivé řádky o zadané maximální délce. Pro jejich uložení je použit jednosměrný spojový seznam. Na rozdíl od použití jednoho dynamicky alokovaného datového pole tak není potřeba znát předem velikost výstupu. Kromě textového obsahu je výstupem ještě počet řádků, na které byl vstupní text rozdělen. Při pozdějších výpočtech tak není potřeba řádky opětovně počítat pomocí průchodu celým spojovým seznamem.

Algoritmus začíná rozdělením vstupního řetězce na jednotlivé části s použitím mezery jako oddělovače. Každá část tvoří jednu položku nově vzniklého spojového seznamu. Algoritmus postupně zpracovává a odstraňuje jednotlivé položky z hlavy tohoto vstupního seznamu tak dlouho, dokud seznam není prázdný. Zároveň vkládá zpracované položky na konec výstupního seznamu. Činnost algoritmu se skládá z následujících kroků:

1. Vložení nové položky, která reprezentuje řádek textu, na konec výstupního spojového seznamu. Jejím obsahem je datové pole znaků s kapacitou odpovídající maximální délce řádku.
2. Přečtení první položky ze vstupního seznamu. Pokud není výstupní datové pole aktuálního řádku prázdné, je před načtenou položku ještě potřeba přidat mezeru. Vleze se tato položka na aktuální řádek?
 - **Ano** – Zapsání textu položky do výstupního datového pole aktuálního řádku. Odstranění první položky vstupního seznamu. Pokračování bodem 2.
 - **Ne** – Pokračování dalším bodem.
3. Vlezla by se položka na nový řádek?
 - **Ano** – Návrat k bodu 1.
 - **Ne** – Pokračování dalším bodem.
4. Položka musí být rozdělena na dvě části. Nyní je potřeba najít pozici, na které má být text rozdělen. Žádná část nesmí být kratší než minimální délka, která je ve výchozím stavu nastavena na 2 znaky. Lze provést rozdělení takovým způsobem, aby první část zabírala celý zbytek řádku a obě části přitom splňovaly minimální délku?
 - **Ano** – Nastavení pozice pro rozdělení na délku zbytku řádku a přechod k bodu 6.
 - **Ne** – Pokračování dalším bodem.

5. První část položky musí být vložena na nový řádek. Pokud by zabrala celý řádek, bude druhá část splňovat minimální délku?
 - **Ano** – Nastavení pozice pro rozdělení na délku celého řádku.
 - **Ne** – Odečtení minimální délky části od celkové délky položky a nastavení pozice pro rozdělení na výsledek tohoto výpočtu.
6. Rozdělení položky na dvě části podle nastavené pozice pro rozdělení. Nahrazení hlavy vstupního seznamu těmito dvěma částmi. Návrat k bodu 2.

Algoritmus je tvořen tak, aby byla preferována dobrá čitelnost před úsporou místa. Nedostatek místa na displayi se totiž týká spíše šířky, takže růst buněk do výšky není problém. Během vykreslování různých rozvrhů hodin s reálnými daty se ukázalo, že rozdělování slov probíhá pouze zřídka. Výukové aktivity se obvykle skládají alespoň ze dvou rozvrhových oken, takže bývá dostatek prostoru na vykreslení většiny slov v celku.

Aby mohl algoritmus správně fungovat, existují v něm dvě omezení, která jsou před začátkem činnosti kontrolována. Maximální délka řádku musí být nastavena na více než 0 znaků. Minimální délka částí, které vznikají dělení slov, musí být menší než maximální délka řádku, ale ne menší než 0.

7.4 Spuštění klientské aplikace

Nejprve je potřeba mít nainstalováno *ESP-IDF* (viz kapitola 4.2). Jedná se o oficiální framework pro vývoj aplikací pro modul ESP32. Poskytuje potřebné knihovny nezbytné ke psaní zdrojového kódu v programovacím jazyce *C*. Také obsahuje skripty sloužící k práci s nástroji, které jsou součástí frameworku. Tyto nástroje slouží především ke kompilaci kódu, nahrávání aplikace na modul ESP32 a monitorování výstupu. K těmto účelům je používán nástroj **make**, který lze spustit s různými volbami, které jsou popsány v následujícím výpisu. Tyto informace platí pro vybranou verzi *ESP-IDF*, pro jiné vydání se postup může mírně lišit. [18]

- **menuconfig** – Otevře menu, ve kterém je k dispozici konfigurace voleb pro vytváření aplikace. V tomto menu je možné měnit různá nastavení, která se týkají přímo modulu ESP32. Jedná se například o možnost deaktivace jednoho z jader procesoru, změna frekvence procesoru nebo nastavení *USB* portu, pomocí kterého je modul ESP32 připojován k počítači. Tento port je vždy potřeba správně nastavit, protože se může mezi různými operačními systémy lišit. Změnit je možné také hodnotu *baud rate*, která určuje rychlost nahrávání aplikace na modul ESP32. Vyšší hodnota znamená větší rychlost, avšak zvolením příliš vysoké rychlosti může nahrávání přestat fungovat.
- **build** – Spustí kompilátor s předdefinovaným nastavením. Kompilaci zdrojového kódu je možné značně urychlit použitím přepínače *-j*, který slouží k vytvoření více současně běžících úloh, takže jsou lépe využity dostupné prostředky počítače.

- **flash** – Nahraje vytvořenou aplikaci na modul ESP32 pomocí připojeného *USB* kabelu. Postup nahrávání je zobrazován v terminálu.
- **monitor** – Resetuje připojený modul ESP32 a zahájí čtení jeho výstupu, který je vypisován do terminálu. Zobrazovány jsou jak informace vypisované uživatelskou aplikací, tak zprávy generované operačním systémem, který na modulu ESP32 běží.

Pro změnu konfigurace aplikace je k dispozici soubor `config.h`. V něm je obsaženo několik konstant, které je potřeba správně nastavit. Mezi nejdůležitější patří přístupové údaje k síti WiFi, adresa serveru pro stahování obsahu k vykreslení, identifikátor, na jehož základě je vrácen obsah ze serveru, délka trvání režimu hibernace mezi opakovaným vykreslováním a informace o připojeném e-ink displayi.

8 Spotřeba elektrické energie

Použitá zařízení, kterými jsou modul ESP32 a e-ink display, by měla být schopna co nejdéle fungovat se zdrojem napájení ve formě baterií. Za tímto účelem je potřeba zjistit, jaká je spotřeba elektrické energie. Ta by měla být co nejnižší nejen když jsou zařízení aktivní, ale také při čekání na zahájení další iterace své činnosti.

Tato kapitola začíná popisem režimů spotřeby elektrické energie, mezi kterými může modul ESP32 přecházet. Následují souhrnné i dílčí informace vycházející z provedených měření spotřeby. Poslední část kapitoly se zabývá výběrem vhodných baterií.

8.1 Režimy spotřeby elektrické energie

Modul ESP32 nabízí několik možností, jak spotřebu elektrické energie omezit. Některé části modulu je možné volitelně vypnout. Také je možné změnit nastavení procesoru. Dostupné volby pro úsporu energie se dají různě kombinovat, takže je možné vytvořit optimální konfiguraci pro každou konkrétní situaci.

Dále jsou vypsány dostupné režimy spotřeby elektrické energie, mezi kterými může modul ESP32 přecházet. Změnu lze provést kdykoliv během činnosti programu. Tyto režimy jsou seřazeny od nejvyšší spotřeby elektrické energie po nejnižší. [28]

- **Aktivní režim** – Všechny části modulu ESP32 jsou napájeny a aktivní, včetně adaptéru pro bezdrátovou komunikaci, takže je možné vytvořit připojení pomocí WiFi nebo Bluetooth. Spotřeba elektrické energie závisí na směru komunikace (přijímání nebo odesílání dat) a také na síle signálu. Čím je dostupný signál slabší, tím je spotřeba energie větší. Odesílání dat pomocí WiFi typicky vyžaduje proud v rozmezí 180 – 240 mA, přičemž příjem vyžaduje pouze 95 – 100 mA. V případě, že je použito Bluetooth, se jedná o 130 mA při odesílání a 95 – 100 mA během přijímání. V této aplikaci je odesílaných informací minimum, jelikož se většina bezdrátové komunikace skládá ze stahování dat k vykreslení. Když je povolena komunikace pomocí WiFi, modul ESP32 automaticky přechází podle potřeby do režimu *spánku modemu* a zpět do *aktivního* režimu, aby bylo dosaženo co největší úspory elektrické energie.
- **Spánek modemu** – V tomto režimu není možné provádět jakoukoliv bezdrátovou komunikaci. Není tedy možné vytvořit připojení pomocí WiFi ani Bluetooth, ostatní funkce modulu ESP32 jsou však nadále k dispozici. Spotřeba elektrické energie závisí především na nastavení procesoru. U něj je možné zvolit jednu ze tří možných frekvencí: 80, 160 nebo 240 MHz. Čím je frekvence vyšší, tím větší je spotřeba elektrické energie. Dalším nastavením procesoru je počet aktivních jader. Nepoužíváním jednoho jádra procesoru může být dosaženo nižší spotřeby elektrické energie, než když jsou obě aktivní. Největší spotřeba se pohybuje v rozsahu 30 – 68 mA při obou aktivních jádrech a frekvenci nastavené

na nejvyšší. Nejúspornější možnost je nastavení nejnižší frekvence a deaktivace jednoho jádra. Spotřeba se pak pohybuje v rozsahu 20 – 25 mA. Spotřeba při ostatních možnostech nastavení se pohybuje v oblasti mezi zmíněnými hodnotami.

- **Lehký spánek** – Činnost procesoru je pozastavena až do probuzení. Ostatní součásti modulu ESP32 zůstávají aktivní. Po probuzení z tohoto režimu spánku je možné pokračovat ve vykonávání programu v místě, ve kterém došlo k usnutí. Odebíraný proud je 0,8 mA.
- **Hluboký spánek** – Součástí modulu ESP32 je RTC paměť a RTC časovač. Jsou to poslední dvě součásti, které v tomto režimu zůstávají napájeny. Volitelně může být ponechán aktivní ještě ULP koprocessor, který lze však také vypnout. V závislosti na něm se odebíraný proud pohybuje v rozsahu 10 – 150 μ A.
- **Hibernace** – Nejúspornější ze všech režimů spánku. Aktivní zůstává pouze RTC časovač a některé piny modulu ESP32. Obojí je možné použít k probuzení z tohoto režimu. Po probuzení začíná vykonávání programu od začátku, stejně jako při prvním připojení ke zdroji napájení. Odebíraný proud je 5 μ A.
- **Vypnutí** – Celý modul ESP32 je vypnut.

Modul ESP32 tráví většinu času v režimu *hibernace*. Použití tohoto nejúspornějšího režimu je možné, protože není potřeba, aby jakákoliv součást modulu zůstala během spánku aktivní. Mimo spánek je úspory elektrické energie dosaženo deaktivací jednoho jádra procesoru a nastavením frekvence na nejnižší.

Když je modul ESP32 zapnut nebo probuzen z *hibernace*, pracuje nejprve střídavě v *aktivním* režimu nebo v režimu *spánku modemu*. V pozdější fázi činnosti modul přechází do *lehkého spánku*, ve kterém čeká na vykreslení obrazu na připojeném e-ink displayi. Poté následuje přechod do režimu *hibernace*.

Kromě pravidelného probouzení modulu ESP32 pomocí časovače, které je opakováno v nastaveném intervalu, je možné nakonfigurovat i různé další zdroje probuzení. Jednou z možností je využití dalšího zařízení, propojení pomocí určitého pinu a použití jako zdroj externího probuzení. Dále je k dispozici použití dotykového senzoru nebo ULP koprocessoru. K probuzení je možné využít také tlačítko, které je dostupné přímo na vývojové desce ESP32. Jeho zmáčknutím dojde k resetování modulu. Je tak možné okamžitě vyvolat nové vykreslení e-ink displaye bez nutnosti čekat na probuzení pomocí časovače. [29]

8.2 Spotřeba modulu ESP32 v různých situacích

Pro představu o spotřebě elektrické energie modulu ESP32 bylo provedeno měření pro různé situace, ve kterých se může modul ocitnout. Tyto naměřené hodnoty jsou k dispozici v tabulce 2. Doplnují a ověřují informace o režimech spotřeby v kapitole 8.1.

Počet jader	Frekvence (MHz)	Odebíraný proud (mA)			
		Práce	Klid	Spánek	Hibernace
1	80	27	20	1	0
2	80	29	22	1	0
1	160	41	27	1	0
2	160	43	29	1	0
1	240	59	35	1	0
2	240	63	40	1	0

Tabulka 2: Odebíraný proud modulem ESP32 v různých situacích

8.2.1 Způsob měření

Pro měření spotřeby elektrické energie byl použit USB měřič připojený mezi počítač a USB kabel vedoucí k modulu ESP32. Na displayi použitého měřiče je zobrazována aktuální hodnota odebíraného proudu s přesností na jednotky mA.

Měření probíhalo bez připojeného e-ink displaye k modulu ESP32 a s vypnutým bezdrátovým adaptérem. Naměřené hodnoty je tak možné použít pro odečet od hodnot naměřených při aktivním bezdrátovém přenosu nebo zapnutém e-ink displayi a zjistit tak, jaký vznikne nárůst spotřeby elektrické energie jejich použitím.

Měření probíhalo pro všechny možnosti, jak lze u modulu ESP32 nastavit frekvenci procesoru a počet aktivních jader. Množství odebíraného proudu bylo sledováno v následujících situacích:

- **Práce** – Umělá zátěž, která je tvořena různými matematickými operacemi v nekonečné smyčce. Cílem je zatížit procesor na maximum.
- **Klid** – Volání funkce `vTaskDelay` s velkou časovou hodnotou v nekonečné smyčce. Modul ESP32 tak tráví veškerý čas čekáním a procesor nevykonává prakticky žádnou činnost.
- **Spánek** – Aktivace režimu lehkého spánku, při kterém je spotřeba elektrické energie nezávislá na nastavení procesoru.
- **Hibernace** – Nejúspornější režim, ve kterém je spotřeba elektrické energie tak malá, že použitým měřicím přístrojem nebyl naměřen žádný proud.

8.2.2 Srovnání výsledků měření

Ve výsledcích měření spotřeby elektrické energie, které jsou zapsány v tabulce 2, lze pozorovat relativně velké rozdíly v odebíraném proudu mezi různými nastaveními frekvence procesoru. Tyto hodnoty se také značně liší mezi klidovým stavem a velkým zatížením procesoru. Režimy spánku a hibernace nejsou při žádném z nastavení procesoru ovlivněny.

Měření odebíraného proudu probíhalo při spuštění jedné úlohy běžící na jednom jádru procesoru. Smyslem je zjistit, zda druhé jádro, které neprovádí žádnou činnost, zvyšuje spotřebu elektrické energie. V naměřených hodnotách lze pozorovat, že při snaze o snížení spotřeby má smysl jedno jádro procesoru deaktivovat.

Podle informací v kapitole 8.1 by se odebíraný proud při zapnutém modulu ESP32 měl pohybovat v rozmezí 20 – 68 mA. V klidovém režimu a nejúspornějším nastavení skutečně naměřená hodnota odpovídá 20 mA, nejvyšší naměřený proud je však 63 mA. To je způsobeno tím, že při měření bylo zatíženo pouze jedno jádro procesoru. Hodnoty pro režim spánku a hibernace odpovídají předpokládaným hodnotám po zaokrouhlení na celé jednotky, jelikož použitá měřicí technika vyšší přesnost nepodporuje.

8.3 Bezdrátový přenos dat

Pro přenos dat mezi serverem a modulem ESP32 je použito připojení pomocí *WiFi*. Další možností, jak by mohlo být navázáno bezdrátové připojení, je použití *Bluetooth*. Pro posouzení vhodnosti jeho použití bylo provedeno přenesení odpovídajícího množství dat, přičemž byla změřena spotřeba elektrické energie.

8.3.1 Vybrané prostředky

Modul ESP32 disponuje nízkoenergetickým a klasickým Bluetooth verze 4.2. Vzhledem k malému množství dat, které má být přeneseno, a také kvůli snaze o dosažení co nejmenší spotřeby elektrické energie, bylo zvoleno nízkoenergetické Bluetooth (*BLE*). Při jeho použití se výstupní výkon během přenosu dat pohybuje v rozmezí od 0,01 mW do 10 mW v závislosti na aktuální situaci. U klasického Bluetooth se tato hodnota může pohybovat až do 100 mW. [7, 30]

Aby byla zajištěna plná kompatibilita mezi oběma komunikujícími stranami, jako druhé zařízení byl zvolen další modul ESP32. Pro připojení a přenos dat mezi těmito moduly byly použity ukázkové projekty `gatt_client` a `gatt_server`, které jsou součástí vývojového frameworku ESP-IDF.

8.3.2 Přenos dat pomocí BLE

Pro simulaci přenosu rozvrhových informací bylo použito 1024 B dat. Před samotným přenosem dat je však potřeba provést ještě některé další kroky. Celý postup, včetně přibližných časových hodnot, je následující:

- Inicializace bezdrátového adaptéru pro použití Bluetooth, ke kterému dochází ihned po zapnutí modulu ESP32, zabere 700 ms.
- Vyhledání serverového modulu klientským je hotovo za 300 ms.
- Vytvoření připojení mezi moduly trvá 1 200 ms.
- Přenos dat z jednoho modulu na druhý proběhne za 300 ms.
- Ukončení připojení je prakticky okamžité.

Naměřené hodnoty odebíraného elektrického proudu jsou následující:

- Nejvyšší spotřeba elektrické energie nastává ihned po zapnutí modulu ESP32 a trvá asi 1 500 ms. Odebíraný proud se pohybuje v rozmezí 60 – 100 mA.
- Dále probíhá odběr 36 mA, který trvá 1 000 ms.
- Následuje vypnutí bezdrátového adaptéru. Při zachování připojení bez odesílání dalších dat by odebíraný proud byl 33 mA.

8.3.3 Připojení pomocí WiFi

Pro porovnání s přenosem dat pomocí Bluetooth je potřeba zmínit, jak vypadá průběh stahování rozvrhových dat pomocí WiFi. Odběr elektrického proudu se po celou dobu pohybuje v rozmezí 60 – 110 mA. Časové informace se mohou při každém spuštění mírně lišit.

- Inicializace WiFi v modulu ESP32 je hotova za 150 ms.
- Připojování k síti WiFi trvá okolo 3 500 ms.
- Stažení dat ze serveru proběhne do 100 ms.
- Vypnutí WiFi v modulu ESP32 zabere 50 ms.

8.3.4 Srovnání WiFi a BLE

Na základě hodnot zjištěných v kapitolách 8.3.2 a 8.3.3 byla sestavena tabulka 3. Tato tabulka obsahuje v prvním sloupci celkový čas, který je potřeba od inicializace bezdrátového spojení, přes stažení dat, až do ukončení spojení. V dalším sloupci je vypočten rozsah hodnot, ve kterých se pohybuje celková spotřeba elektrické energie. Poslední sloupec tabulky obsahuje spotřebu elektrické energie, kterou tvoří provoz samotného rádiového spojení. Tyto hodnoty jsou vypočítány odečtením spotřeby elektrické energie modulu ESP32 bez zapnutého bezdrátového adaptéru.

Oba typy připojení tráví většinu času inicializací a připojováním, samotný přenos dat trvá jen velmi krátkou dobu. Potřebná doba se však může v jiných podmínkách změnit, například při použití jiného zařízení, se kterým je spojení vytvářeno nebo při jiné síle a kvalitě signálu.

V tomto případě je použití nízkoenergetického Bluetooth rychlejší a méně energeticky náročné. Je to však dáno tím, že je přenášeno pouze malé množství dat. Naměřené hodnoty vznikly při přenosu dat rozvrhu hodin o velikosti 1 KiB, zatímco velikost obrazové bitmapy je 90 KiB. Doba přenosu většího objemu dat pomocí BLE se přímo úměrně zvětší, při stahování pomocí WiFi však doba příliš nenaroste. Je to způsobeno tím, že u WiFi je z naměřených 100 ms komunikace pouze část této doby použita pro stahování požadovaných dat, zbytek času trvá vytvoření spojení se serverem. Přenos dat pomocí BLE probíhá přes již vytvořené spojení.

	Potřebný čas (ms)	Spotřeba (mAs)	
		Celkem	Rádío
WiFi	3 800	228 – 418	125 – 342
BLE	2 500	126 – 186	59 – 136

Tabulka 3: Srovnání spotřeby elektrické energie u WiFi a BLE

8.4 Vykreslení obsahu na e-ink display

Po stažení obrazových dat je možné začít s vykreslováním obsahu na připojený e-ink display. Celý proces vykreslování se skládá z následujících fází:

- Aktivace displaye, krátké vyčkání na jeho zapnutí a poté přenos 180 KiB obrazových dat. Tato fáze trvá 5 260 ms. Přestože už je display zapnut, většinu spotřeby elektrické energie tvoří modul ESP32, protože display ještě nezačal s vykreslováním.
- Přechod modulu ESP32 do režimu lehkého spánku. Napájení displaye zůstává připojeno a začíná vykreslování obsahu, které trvá 41 000 ms. Po tuto dobu tvoří většinu spotřeby elektrické energie display, zatímco spotřeba modulu ESP32 je téměř nulová.
- Probuzení modulu ESP32 z režimu lehkého spánku. Následuje vypnutí displaye, které proběhne téměř okamžitě.

Na základě zjištěných informací byla vytvořena tabulka 4, která obsahuje potřebnou dobu, odebíraný proud a celkovou spotřebu elektrické energie ve dvou fázích činnosti e-ink displaye. Pro fázi vykreslení je k dispozici také řádek s předpokládanými hodnotami, které jsou převzaty z kapitoly 3.3.

	Trvání (ms)	Proud (mA)	Spotřeba (mAs)
Inicializace displaye	5 260	26 – 36	136,8 – 189,4
Vykreslení (naměřeno)	41 000	38 – 58	1 558 – 2 378
Vykreslení (předpokládáno)	41 000	24,4 – 71,3	1 000,4 – 2 923,3

Tabulka 4: Spotřeba elektrické energie v jednotlivých fázích vykreslování obsahu

Za předpokladu, že samotný modul ESP32 odebírá v první fázi činnosti e-ink displaye proud o velikosti 20 – 27 mA a ve druhé fázi 1 mA (viz tabulka 2), lze společně s hodnotami v tabulce 4 určit, jakou spotřebu má e-ink display a jakou modul ESP32. Tyto hodnoty jsou k dispozici v tabulce 5.

	Trvání (ms)	Zařízení	Proud (mA)	Spotřeba (mAs)
Inicializace	5 260	ESP32	20 – 27	105 – 142
		Display	0 – 16	0 – 84
Vykreslení	41 000	ESP32	1	41
		Display	37 – 57	1 517 – 2 327
Celkem	46 260	ESP32	—	146 – 183
		Display	—	1 517 – 2 421

Tabulka 5: Rozdělení spotřeby elektrické energie při vykreslování obsahu

8.5 Souhrnné informace o spotřebě elektrické energie

Jednotlivé získané informace o spotřebě elektrické energie mohou být použity k vytvoření souhrnných informací, které se týkají celého procesu vykreslení obsahu na e-ink display. Tyto informace jsou dostupné v následujících podkapitolách.

8.5.1 Celková spotřeba

S použitím hodnot, které byly naměřeny a vypočteny v kapitolách 8.3 a 8.4, je možné určit celkové množství elektrické energie, která je spotřebována při jednom stažení a vykreslení obsahu na e-ink display. Spolu s celkovou dobou, která je pro vykonání potřeba, jsou tyto informace k dispozici v tabulce 6.

	Trvání (ms)	Spotřeba (mAs)
Stažení dat (WiFi)	3 800	228 – 418
Vykreslení obsahu	46 260	1 663 – 2 604
Celkem	50 060	1 891 – 3 022

Tabulka 6: Celková spotřeba elektrické energie

8.5.2 Rozdělení spotřeby na modul ESP32 a e-ink display

Kromě celkové spotřeby elektrické energie je užitečnou informací také to, jakou část spotřeby tvoří modul ESP32 a jakou e-ink display. Tuto informaci obsahuje tabulka 7, která je založena opět na datech z předchozích kapitol.

	Spotřeba (mAs)	Podíl (%)
ESP32	374 – 601	25
Display	1 517 – 2 421	75
Celkem	1 891 – 3 022	100

Tabulka 7: Rozdělení celkové spotřeby elektrické energie

8.6 Zhodnocení spotřeby elektrické energie

Závěrem měření spotřeby elektrické energie je zhodnocení získaných hodnot. Ty jsou v následujících podkapitolách použity ke zjištění možné úspory elektrické energie a pro odhad předpokládané doby provozu modulu ESP32 a e-ink displaye na baterie.

8.6.1 Možnosti úspory elektrické energie

Pro zjištění možné úspory elektrické energie, která by mohla nastat použitím nízkoenergetického Bluetooth místo WiFi, byla vytvořena tabulka 8. První sloupec zobrazuje možnou úsporu pouze pro fázi stahování dat, zatímco druhý sloupec zachycuje úsporu v kontextu celkové spotřeby elektrické energie.

	Spotřeba elektrické energie (mAs)	
	Pouze stažení dat	Včetně vykreslení
WiFi	228 – 418	1 891 – 3 022
BLE	126 – 186	1 789 – 2 790
Rozdíl	102 – 232	102 – 232
Úspora	45 – 56 %	5 – 8 %

Tabulka 8: Možná úspora elektrické energie při použití BLE místo WiFi

Množství elektrické energie, která je spotřebována pro stažení dat, je možné použitím BLE snížit přibližně na polovinu. V rámci celého procesu vykreslení obsahu na e-ink display je však tato úspora méně významná, jelikož je celková spotřeba elektrické energie snížena pouze o 5 – 8 %. Tato úspora se týká pouze přenášení rozvrhových dat, jelikož mají malou velikost. Pro vykreslování webových stránek je vhodnější použití WiFi, protože je přenášena mnohonásobně větší obrazová bitmapa (viz kapitola 8.3.4).

Pro případné použití Bluetooth je vhodné zvážit plánované umístění modulu ESP32. V některých situacích může být pokrytí oblasti signálem Bluetooth problematické, přičemž přítomnost signálu WiFi obvykle bývá dostupnější.

Další možností, jak snížit energetickou náročnost, by mohl být výběr jiného e-ink displaye. Právě používaný display totiž stojí za 75 % celkové spotřeby elektrické energie (viz tabulka 7). Úspornější e-ink display, který splňuje všechny požadavky (viz kapitola 3.3), však v dnešní době nemusí být vůbec k dispozici.

8.6.2 Doba provozu na baterie

Pro předpoklad doby provozu modulu ESP32 a e-ink displaye připojených na baterie je možné použít data o spotřebě získaná provedeným měřením v předchozích kapitolách. Vydělením celkové kapacity použitých baterií a spotřeby elektrické energie při jednom překreslení lze vypočítat celkový počet vykreslení, než dojde k vybití baterií.

Celková spotřeba elektrické energie při jednom překreslení obsahu se pohybuje v hodnotách 1 891 – 3 022 mAs (viz tabulka 6), což odpovídá přibližně 0,53 – 0,84 mAh. Pokud by například byly použity baterie s celkovou kapacitou 5 000 mAh, bylo by teoreticky možné provést 5 956 – 9 518 překreslení. Při dvou překresleních za den lze dosáhnout doby provozu 8 – 13 let. Kapacitu baterií lze vypotřebovat za 1 rok, pokud bude překreslování probíhat 16 – 26 krát denně. Při reálném nasazení však lze očekávat mírnou odchylku oproti těmto hodnotám. Nebyly totiž zahrnuty okolnosti, jako například možná odlišnost v kvalitě WiFi signálu nebo samovybití baterií.

Při výpočtu nebyla zahrnuta spotřeba modulu ESP32 během nečinnosti, při které modul setrvává v režimu *hibernace* (viz kapitola 8.1). Tuto spotřebu je však vzhledem k odebíranému proudu o velikosti pouhých 5 μ A možné zanedbat. Výraznější dopad na výdrž baterií má spotřeba vypnutého e-ink displaye. Kvůli odebíranému proudu, který je téměř 4 mA, je zkrácena předpokládaná výdrž na necelé 2 měsíce. Tento nedostatek by bylo možné vyřešit použitím *MOSFET* tranzistoru, pomocí kterého by byl e-ink display připojen k napájení pouze po dobu vykreslování a poté by byl odpojen.

8.7 Výběr vhodných baterií

Existuje celá řada různých druhů baterií, které je možné použít jako zdroj napájení. Při výběru je vhodné zvážit následující vlastnosti:

- **Typ** – Odlišné typy baterií mohou mít různý tvar, rozměry a hmotnost. To může být důležité například při výrobě obalu, do kterého by byl umístěn e-ink display společně s modulem ESP32 a bateriemi.
- **Napětí** – Vzhledem k provoznímu napětí e-ink displaye a modulu ESP32 je potřeba zvolit baterie, které poskytují celkové napětí o velikosti 3 V. Přestože je možné použít jednu baterii s tímto napětím, typicky jsou dostupnější variantou dvě baterie s napětím 1,5 V.
- **Kapacita** – Různé baterie pracující na odlišném principu mohou disponovat jinou kapacitou. Kromě výběru vhodného druhu je možné zajistit vyšší kapacitu pořízením baterií o větších rozměrech.
- **Samovybití** – Baterie časem ztrácejí využitelnou kapacitu, i když nejsou právě používány. Pro dlouhodobé fungování je nutné, aby byl použit druh baterií s co nejmenším samovybitím.

- **Vybíjecí charakteristika** – Používáním baterie klesá napětí, které je baterie schopna poskytovat. V určité fázi vybíjení klesne napětí natolik, že připojený modul ESP32 a e-ink display nebudou dále moci fungovat a zbývající kapacita baterie zůstane nevyužita. Vhodné je vybrat takový druh baterie, který tímto poklesem napětí trpí co nejméně.
- **Možnost dobíjení** – Kromě baterií na jedno použití existují také druhy, které je po vybití možné znovu nabít. U těchto druhů se mohou vyskytovat některé horší vlastnosti a vyšší cena. Díky možnosti opakovaného používání má jejich použití smysl především při častém vybíjení.
- **Cena** – Mezi různými druhy baterií může být mnohonásobný rozdíl v ceně, přičemž požadované vlastnosti mohou zůstat podobné. Jedinou ze zmíněných vlastností, která pro správné fungování musí stále zůstat stejná, je odpovídající hodnota napětí.

9 Závěr

V této diplomové práci bylo navázáno na semestrální projekt, ve kterém bylo zprovozněno zobrazování statických snímků webových stránek pomocí vybraného e-ink displaye a vývojové desky ESP32. Projekt byl rozšířen o možnost vykreslování rozvrhů hodin místností. Byl vytvořen postup, pomocí kterého je na základě textových informací o výuce vytvářen obraz, který obsahuje tabulku s rozvrhem hodin. Vykreslování bylo optimalizováno pro vybraný e-ink display, aby bylo dosaženo co nejlepší čitelnosti, přehlednosti a využití prostoru na displayi.

Práce byla rozdělena na dvě hlavní části. První část je server, který slouží k zobrazování přehledů, konfiguraci položek k vykreslování a poskytování těchto položek ke stažení. Také pořízuje snímky webových stránek pomocí webového prohlížeče a získává rozvrhy hodin z univerzitního informačního systému pomocí rozhraní *INBUS*. Druhou částí je klient, jehož úkolem je v pravidelných intervalech stahovat ze serveru data k vykreslení na připojený e-ink display. Klientů může existovat více současně. Na základě přiřazených identifikátorů je jim předáván nakonfigurovaný obsah k vykreslení, který je na serveru předem připraven ve vhodném formátu.

Vývoj aplikace běžící na klientském zařízení byl zaměřen na maximální úsporu elektrické energie. K tomu účelu byly využity dostupné možnosti použitého modulu ESP32, kterými jsou režimy spánku, vypínání bezdrátového adaptéru a nastavení procesoru. Pro ověření energetické náročnosti byla provedena řada měření, která poskytují informace o jednotlivých fázích probíhajícího procesu a o spotřebě elektrické energie samotného e-ink displaye a modulu ESP32. Byl změřen rozdíl mezi použitím *Bluetooth* a *WiFi*, ze kterého vyplynulo, že by v případě použití *Bluetooth* mohlo dojít pouze k mírnému snížení energetické náročnosti. Na základě spotřeby elektrické energie při jednom vykreslení obsahu byla odhadnuta možná doba provozu na baterie na několik let. Vzhledem k neustálému mírnému odběru proudu vypnutého e-ink displaye je však tato doba provozu, bez provedení dalších hardwarových úprav, zkrácena na několik týdnů.

Vzhledem k relativně vysoké spotřebě elektrické energie použitého e-ink displaye by mohlo být v budoucnu vhodné prozkoumat, zda není dostupný vhodnější model. Kromě nižší energetické náročnosti by mohlo být k dispozici například jemnější rozlišení nebo lepší barevné možnosti. Práce by pak mohla být rozšířena o podporu jednoho nebo více dalších e-ink displayů. Místo vybrané vývojové desky ESP32 by bylo možné použít jiné zařízení, které by mělo menší energetickou náročnost nebo by umožňovalo použití jiného typu bezdrátového přenosu. Při pokračování ve vývoji tohoto projektu by mohly být přidány další funkce, jako například plánovač aktualizací obsahu, aby aktualizace nemusely probíhat pouze v pravidelných intervalech. Také by mohlo být vykreslování rozvrhů hodin rozšířeno o zobrazování dalších informací, jako je například podrobnější popis místnosti nebo volitelná textová zpráva.

Vytvořený projekt je připraven k reálnému používání na univerzitě, kde by mohly být nahrazeny nynější papírové rozvrhy hodiny u místností, které jsou hojně používány. Díky možnosti vykreslování webových stránek je však možné nalézt i další uplatnění na místech, kde by zobrazování informací pomocí e-ink displaye mělo smysl.

Literatura

1. *Amazon: All-new Kindle – White* [online]. 2019 [cit. 2020-04-25]. Dostupné z: <https://www.amazon.com/All-new-Kindle-Built-Unlimited-auto-renewal/dp/B085VX2RKH>.
2. *Amazon: Lexar JumpDrive M10 128GB Secure USB 3.0 flash drive LJDM10-128BSBNA* [online]. 2013 [cit. 2020-04-25]. Dostupné z: <https://www.amazon.com/Lexar-JumpDrive-128GB-Secure-LJDM10-128BSBNA/dp/B00DGHNY2U>.
3. *W3Techs: Usage statistics of CSS for websites* [online]. 2020 [cit. 2020-03-23]. Dostupné z: <https://w3techs.com/technologies/details/ce-css>.
4. *W3Techs: Usage statistics of JavaScript as client-side programming language on websites* [online]. 2020 [cit. 2020-03-23]. Dostupné z: <https://w3techs.com/technologies/details/cp-javascript>.
5. *ESP-IDF Programming Guide: Wi-Fi Driver* [online]. 2020 [cit. 2020-03-23]. Dostupné z: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/wifi.html>.
6. *ESP32: Technical Reference Manual* [online]. 2019 [cit. 2020-03-23]. Dostupné z: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf.
7. *ESP32-WROOM-32: Datasheet* [online]. 2019 [cit. 2020-03-23]. Dostupné z: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf.
8. *Product Specifications: 12" 3-Colors TFT EPD Panel* [online]. 2018 [cit. 2020-03-23]. Dostupné z: https://www.pervasivedisplays.com/wp-content/uploads/2019/06/1p1xx-00_03_E2B98FS081_20180502.pdf.
9. *E Ink: Electronic Ink* [online]. 2017 [cit. 2020-03-23]. Dostupné z: <https://www.eink.com/electronic-ink.html>.
10. *User's Guide: EPD Extension Kit Gen2* [online]. 2019 [cit. 2020-04-28]. Dostupné z: https://www.pervasivedisplays.com/wp-content/uploads/2019/06/EPD-Extension-Kit-Gen2-EXT2_User-Guide_Rev08_20191111.pdf.
11. *Circuit Basics: Basics of the SPI Communication Protocol* [online]. 2016 [cit. 2020-03-23]. Dostupné z: <https://www.circuitbasics.com/basics-of-the-spi-communication-protocol/>.
12. *pip: The Python Package Installer* [online]. 2020 [cit. 2020-03-23]. Dostupné z: <https://pip.pypa.io/en/stable/>.
13. *venv: Creation of virtual environments* [online]. 2020 [cit. 2020-03-23]. Dostupné z: <https://docs.python.org/3/library/venv.html>.

14. *GitHub: CEF Python* [online]. 2020 [cit. 2020-03-23]. Dostupné z: <https://github.com/cztomczak/cefpython/>.
15. *Read the Docs: Pillow* [online]. 2019 [cit. 2020-03-23]. Dostupné z: <https://pillow.readthedocs.io/en/stable/index.html>.
16. *Django: How to get Django* [online]. 2020 [cit. 2020-03-23]. Dostupné z: <https://www.djangoproject.com/download/>.
17. *ESP-IDF Programming Guide: Standard Setup of Toolchain for Linux* [online]. 2019 [cit. 2020-03-23]. Dostupné z: <https://docs.espressif.com/projects/esp-idf/en/v3.3.1/get-started/linux-setup.html>.
18. *ESP-IDF Programming Guide: Get Started* [online]. 2018 [cit. 2020-03-23]. Dostupné z: <https://docs.espressif.com/projects/esp-idf/en/v3.3.1/get-started/index.html>.
19. *ESP-IDF Programming Guide: Add IDF_PATH to User Profile* [online]. 2018 [cit. 2020-03-23]. Dostupné z: https://docs.espressif.com/projects/esp-idf/en/v3.3.1/get-started/add-idf_path-to-profile.html.
20. *Django: Overview* [online]. 2020 [cit. 2020-03-23]. Dostupné z: <https://www.djangoproject.com/start/overview/>.
21. *Bitbucket: Chromium Embedded Framework* [online]. 2018 [cit. 2020-03-23]. Dostupné z: <https://bitbucket.org/chromiumembedded/cef/src/master/>.
22. *The Chromium Projects: Chromium* [online] [cit. 2020-03-23]. Dostupné z: <https://www.chromium.org/Home>.
23. *GitHub: CEF Python Tutorial – Off-screen rendering* [online]. 2018 [cit. 2020-03-23]. Dostupné z: <https://github.com/cztomczak/cefpython/blob/master/docs/Tutorial.md#off-screen-rendering>.
24. *Read the Docs: Pillow Image Module – Image.convert* [online]. 2019 [cit. 2020-03-23]. Dostupné z: <https://pillow.readthedocs.io/en/stable/reference/Image.html#PIL.Image.Image.convert>.
25. *RFC 4122: A Universally Unique Identifier (UUID) URN Namespace* [online]. 2005 [cit. 2020-03-23]. Dostupné z: <https://tools.ietf.org/html/rfc4122>.
26. *Application Note: 12" Spectra EPD with the cascaded G2.1 iTC* [online]. 2017 [cit. 2020-03-23]. Dostupné z: https://www.pervasivedisplays.com/wp-content/uploads/2019/06/ApplicationNote_EPD1200_Spectra_with_G2.1_171106.pdf.
27. *ESP-IDF Programming Guide: Logging library* [online]. 2020 [cit. 2020-03-23]. Dostupné z: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/log.html>.

28. *ESP32 Series: Datasheet* [online]. 2020 [cit. 2020-03-23]. Dostupné z: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.
29. *ESP-IDF Programming Guide: Sleep Modes* [online]. 2020 [cit. 2020-03-23]. Dostupné z: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/sleep_modes.html.
30. *Specification of the Bluetooth System: Version 4.2* [online]. 2014 [cit. 2020-03-23]. Dostupné z: https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=441541.